图灵程序设计丛书

LIONlab Trento University

The LION Way

Machine Learning plus Intelligent Optimization

机器学习与优化

[意] 罗伯托 · 巴蒂蒂　毛罗 · 布鲁纳托　著

王彧弋　译

- 摒弃复杂的公式推导，从实践上手机器学习
- 人工智能领域先驱、IEEE会士巴蒂蒂教授领导的LION实验室多年机器学习经验总结

中国工信出版集团　人民邮电出版社 POSTS & TELECOM PRESS

作者简介

罗伯托·巴蒂蒂（Roberto Battiti）

人工智能领域先驱，IEEE会士。因在无功搜索优化（RSO）方向做出了开创性的工作而名震学界。目前为意大利特伦托大学教授，同时担任特伦托大学机器学习与智能优化实验室（LION lab）主任。

毛罗·布鲁纳托（Mauro Brunato）

意大利特伦托大学助理教授，LION研究团队成员。

译者简介

王彧弋

博士，现于瑞士苏黎世联邦理工学院从事研究工作，主要研究方向为理论计算机科学与机器学习。

封面设计 | 83720326@qq.com
自在书装设计

数字版权声明

图灵社区的电子书没有采用专有客户端，您可以在任意设备上，用自己喜欢的浏览器和PDF阅读器进行阅读。

但您购买的电子书仅供您个人使用，未经授权，不得进行传播。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

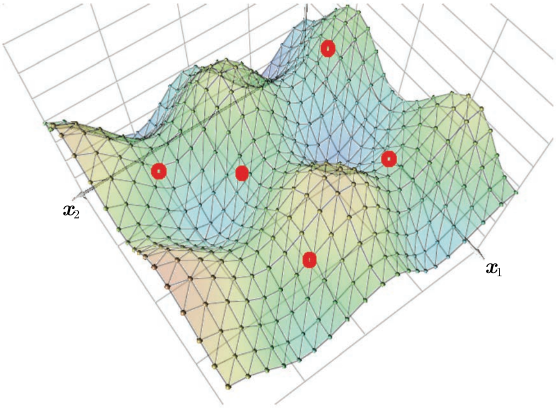


图 1-2　从样本中使用克里金法构造模型。一些样本在图中用点标示出来。表面的高度和颜色依赖于产金量

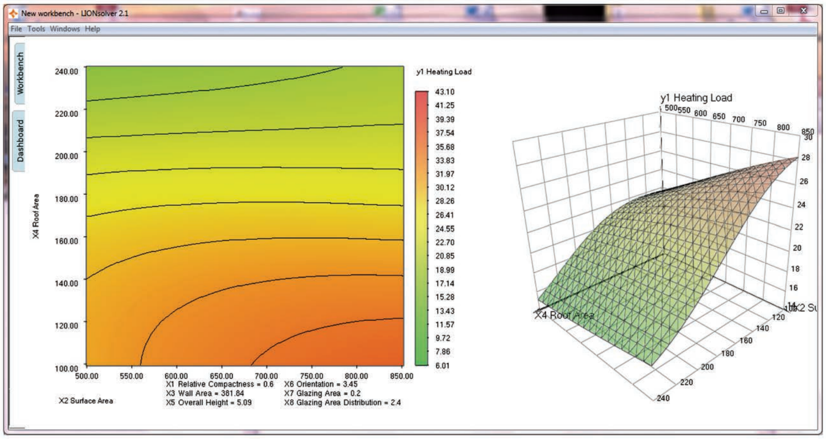


图 9-4　用 LION 软件 Sweeper 分析神经网络的输出。输出值和冬季加热房子消耗的能量，是输入参数的函数。图中展示了颜色编码的输出（左）和表面图（右）。非线性是可见的

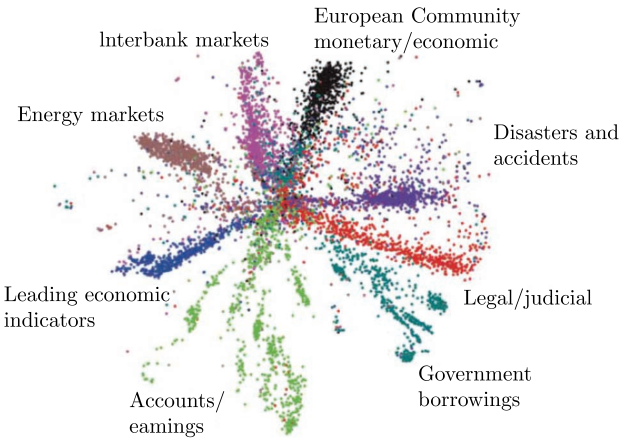


图 10-4　代码由一个 2000-500-250-125-2 自编码器根据路透社的新闻故事生成。图中用不同的颜色对应于不同主题的聚类，这是清晰可见的（详见参考文献 [57]）

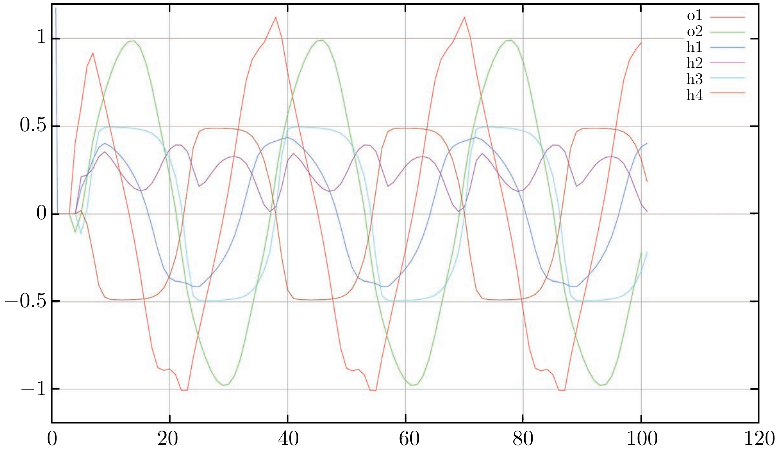


图 14-3　递归神经网络沿着环形训练：输出和隐藏层神经元

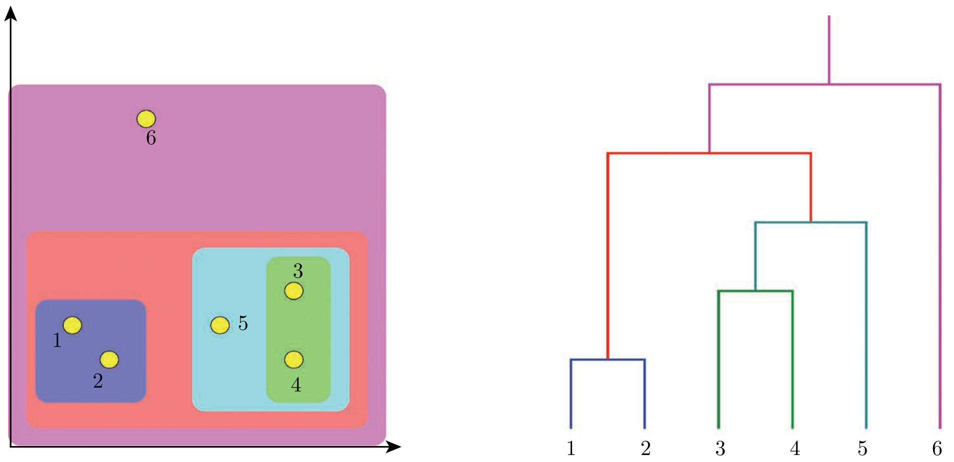


图 16-1　二维空间中数据点自底向上聚类示意图（使用标准欧几里得距离），每个数据点都由两个数值构成

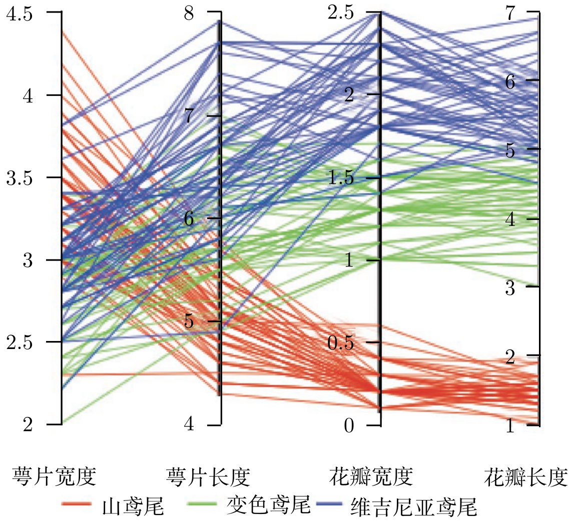


图 16-5　费希尔鸢尾花数据集（每朵花包含 4 个度量属性）的平行坐标展示，每个属性都用一个垂直轴表示，数据中的第 i 项属性值表示为折线与对应的第 i 个垂直轴的交点

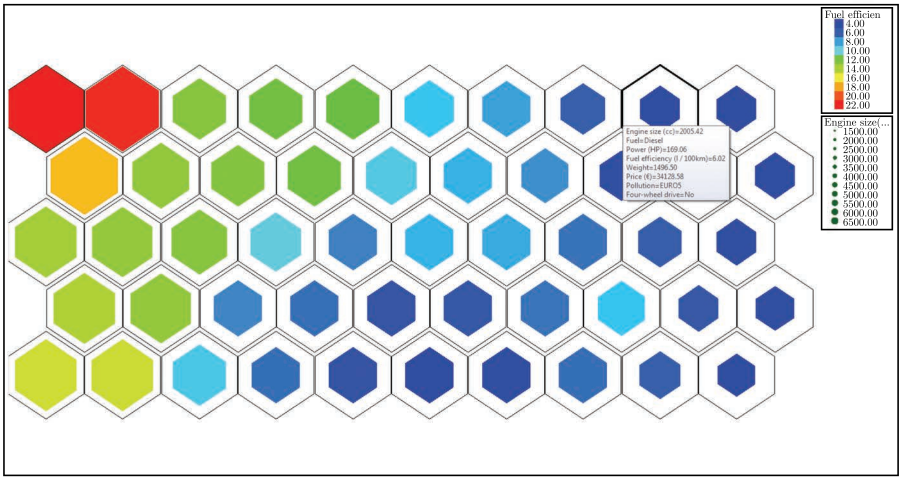


图 17-5　一个 SOM，颜色和大小取决于二维原型向量的两个坐标，可以将鼠标移到神经元上来显示原型的值（通过 LIONoso.org 提供的软件）

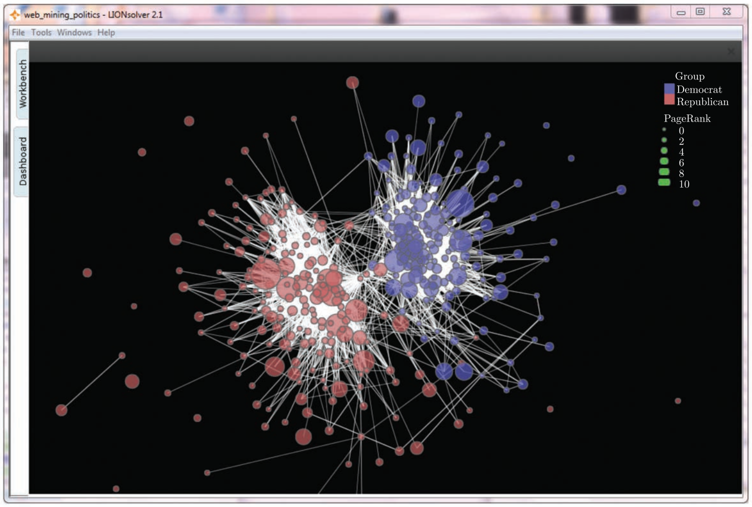


图 19-2　社交网络分析：美国议员的可视化网络。两个政党（从聚类软件无法得到）呈现出非常不同的两个类别

TURING 图灵程序设计丛书

The LION Way

Machine Learning plus Intelligent Optimization

机器学习与优化

【意】Roberto Battiti　Mauro Brunato　著
王彧弋　译

人民邮电出版社
北　京

图书在版编目（CIP）数据

机器学习与优化 /（意）罗伯托·巴蒂蒂
（Roberto Battiti），（意）毛罗·布鲁纳托
（Mauro Brunato）著；王彧弋译. —北京：人民邮电
出版社，2018. 5
（图灵程序设计丛书）
ISBN 978-7-115-48029-3

Ⅰ. ①机… Ⅱ. ①罗… ②毛… ③王… Ⅲ. ①机器学
习 Ⅳ. ①TP181

中国版本图书馆 CIP 数据核字（2018）第 044097 号

内 容 提 要

本书是机器学习实战领域的一本佳作，从机器学习的基本概念讲起，旨在将初学者引入机器学习的大门，并走上实践的道路。本书通过讲解机器学习中的监督学习和无监督学习，并结合特征选择和排序、聚类方法、文本和网页挖掘等热点问题，论证了“优化是力量之源”这一观点，为机器学习在企业中的应用提供了切实可行的操作建议。

本书适合从事机器学习领域工作的相关人员，以及任何对机器学习感兴趣的读者。

◆ 著　　　　〔意〕罗伯托·巴蒂蒂　毛罗·布鲁纳托
译　　　　王彧弋
责任编辑　朱　巍
执行编辑　温　雪　黄志斌
责任印制　周昇亮

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京　　　　　　印刷

◆ 开本：800×1000　1/16
印张：17.5　　　　彩插：2
字数：420 千字　　　　2018 年 5 月第 1 版
印数：1-3 500 册　　　　2018 年 5 月北京第 1 次印刷

著作权合同登记号　图字：01-2014-4553 号

定价：89.00 元

读者服务热线：(010)51095186 转 600　印装质量热线：(010) 81055316

反盗版热线：(010)81055315

版 权 声 明

目　录

第 1 章　引言 · · · · · · 1
1.1　学习与智能优化：燎原之火 · · · · · · 1
1.2　寻找黄金和寻找伴侣 · · · · · · 3
1.3　需要的只是数据 · · · · · · 5
1.4　超越传统的商业智能 · · · · · · 5
1.5　LION 方法的实施 · · · · · · 6
1.6　“动手”的方法 · · · · · · 6
第 2 章　懒惰学习：最近邻方法 · · · · · · 9
第 3 章　学习需要方法 · · · · · · 14
3.1　从已标记的案例中学习：最小化和泛化 · · · · · · 16
3.2　学习、验证、测试 · · · · · · 18
3.3　不同类型的误差 · · · · · · 21

第一部分　监督学习

第 4 章　线性模型 · · · · · · 26
4.1　线性回归 · · · · · · 27
4.2　处理非线性函数关系的技巧 · · · · · · 28
4.3　用于分类的线性模型 · · · · · · 29
4.4　大脑是如何工作的 · · · · · · 30
4.5　线性模型为何普遍，为何成功 · · · · · · 31
4.6　最小化平方误差和 · · · · · · 32
4.7　数值不稳定性和岭回归 · · · · · · 34
第 5 章　广义线性最小二乘法 · · · · · · 37
5.1　拟合的优劣和卡方分布 · · · · · · 38
5.2　最小二乘法与最大似然估计 · · · · · · 42
5.2.1　假设检验 · · · · · · 42
5.2.2　交叉验证 · · · · · · 44
5.3　置信度的自助法 · · · · · · 44
第 6 章　规则、决策树和森林 · · · · · · 50
6.1　构造决策树 · · · · · · 52
6.2　民主与决策森林 · · · · · · 56
第 7 章　特征排序及选择 · · · · · · 59
7.1　特征选择：情境 · · · · · · 60
7.2　相关系数 · · · · · · 62
7.3　相关比 · · · · · · 63
7.4　卡方检验拒绝统计独立性 · · · · · · 64
7.5　熵和互信息 · · · · · · 64
第 8 章　特定非线性模型 · · · · · · 67
8.1　logistic 回归 · · · · · · 67
8.2　局部加权回归 · · · · · · 69
8.3　用 LASSO 来缩小系数和选择输入值 · · · · · · 72
第 9 章　神经网络：多层感知器 · · · · · · 76
9.1　多层感知器 · · · · · · 78
9.2　通过反向传播法学习 · · · · · · 80
9.2.1　批量和 bold driver 反向传播法 · · · · · · 81
9.2.2　在线或随机反向传播 · · · · · · 82
9.2.3　训练多层感知器的高级优化 · · · · · · 83
第 10 章　深度和卷积网络 · · · · · · 84
10.1　深度神经网络 · · · · · · 85
10.1.1　自动编码器 · · · · · · 86
10.1.2　随机噪声、屏蔽和课程 · · · · · · 88
10.2　局部感受野和卷积网络 · · · · · · 89
第 11 章　统计学习理论和支持向量机 · · · · · · 94
11.1　经验风险最小化 · · · · · · 96
11.1.1　线性可分问题 · · · · · · 98
11.1.2　不可分问题 · · · · · · 100
11.1.3　非线性假设 · · · · · · 100
11.1.4　用于回归的支持向量 · · · · · · 101
第 12 章　最小二乘法和健壮内核机器 · · · · · · 103
12.1　最小二乘支持向量机分类器 · · · · · · 104

12.2 健壮加权最小二乘支持向量机 …… 106
12.3 通过修剪恢复稀疏 …… 107
12.4 算法改进：调谐 QP、原始版本、无补偿 …… 108
第 13 章 机器学习中的民主 …… 110
13.1 堆叠和融合 …… 111
13.2 实例操作带来的多样性：装袋法和提升法 …… 113
13.3 特征操作带来的多样性 …… 114
13.4 输出值操作带来的多样性：纠错码 …… 115
13.5 训练阶段随机性带来的多样性 …… 115
13.6 加性 logistic 回归 …… 115
13.7 民主有助于准确率–拒绝的折中 …… 118
第 14 章 递归神经网络和储备池计算 …… 121
14.1 递归神经网络 …… 122
14.2 能量极小化霍普菲尔德网络 …… 124
14.3 递归神经网络和时序反向传播 …… 126
14.4 递归神经网络储备池学习 …… 127
14.5 超限学习机 …… 128

第二部分 无监督学习和聚类

第 15 章 自顶向下的聚类：K 均值 …… 132
15.1 无监督学习的方法 …… 134
15.2 聚类：表示与度量 …… 135
15.3 硬聚类或软聚类的 K 均值方法 …… 137
第 16 章 自底向上（凝聚）聚类 …… 142
16.1 合并标准以及树状图 …… 142
16.2 适应点的分布距离：马氏距离 …… 144
16.3 附录：聚类的可视化 …… 146
第 17 章 自组织映射 …… 149
17.1 将实体映射到原型的人工皮层 …… 150
17.2 使用成熟的自组织映射进行分类 …… 153
第 18 章 通过线性变换降维（投影） …… 155
18.1 线性投影 …… 156
18.2 主成分分析 …… 158
18.3 加权主成分分析：结合坐标和关系 …… 160
18.4 通过比值优化进行线性判别 …… 161
18.5 费希尔线性判别分析 …… 163
第 19 章 通过非线性映射可视化图与网络 …… 165
19.1 最小应力可视化 …… 166
19.2 一维情况：谱图绘制 …… 168
19.3 复杂图形分布标准 …… 170
第 20 章 半监督学习 …… 174
20.1 用部分无监督数据进行学习 …… 175
20.1.1 低密度区域中的分离 …… 177
20.1.2 基于图的算法 …… 177
20.1.3 学习度量 …… 179
20.1.4 集成约束和度量学习 …… 179

第三部分 优化：力量之源

第 21 章 自动改进的局部方法 …… 184
21.1 优化和学习 …… 185
21.2 基于导数技术的一维情况 …… 186
21.2.1 导数可以由割线近似 …… 190
21.2.2 一维最小化 …… 191
21.3 求解高维模型（二次正定型） …… 191
21.3.1 梯度与最速下降法 …… 194
21.3.2 共轭梯度法 …… 196
21.4 高维中的非线性优化 …… 196
21.4.1 通过线性查找的全局收敛 …… 197
21.4.2 解决不定黑塞矩阵 …… 198
21.4.3 与模型信赖域方法的关系 …… 199
21.4.4 割线法 …… 200
21.4.5 缩小差距：二阶方法与线性复杂度 …… 201
21.5 不涉及导数的技术：反馈仿射振荡器 …… 202
21.5.1 RAS：抽样区域的适应性 …… 203
21.5.2 为健壮性和多样化所做的重复 …… 205

第 22 章　局部搜索和反馈搜索优化 · · · · · 211
22.1　基于扰动的局部搜索 · · · · · · · · · · · · · · · 212
22.2　反馈搜索优化：搜索时学习 · · · · · · · · · 215
22.3　基于禁忌的反馈搜索优化 · · · · · · · · · · · 217
第 23 章　合作反馈搜索优化 · · · · · · · · · · · · · 222
23.1　局部搜索过程的智能协作 · · · · · · · · · · · 223
23.2　CoRSO：一个政治上的类比 · · · · · · · · 224
23.3　CoRSO 的例子：RSO 与 RAS 合作 · 226
第 24 章　多目标反馈搜索优化 · · · · · · · · · · · 232
24.1　多目标优化和帕累托最优 · · · · · · · · · · · 233
24.2　脑–计算机优化：循环中的用户 · · · · · 235

第四部分　应用精选

第 25 章　文本和网页挖掘 · · · · · · · · · · · · · · · 240
25.1　网页信息检索与组织 · · · · · · · · · · · · · · · 241
25.1.1　爬虫 · 241
25.1.2　索引 · 242
25.2　信息检索与排名 · · · · · · · · · · · · · · · · · · · 244
25.2.1　从文档到向量：向量–空间模型 · 245
25.2.2　相关反馈 · 247
25.2.3　更复杂的相似性度量 · · · · · · 248
25.3　使用超链接来进行网页排名 · · · · · · · · 250
25.4　确定中心和权威：HITS · · · · · · · · · · · · 254
25.5　聚类 · 256
第 26 章　协同过滤和推荐 · · · · · · · · · · · · · · · 257
26.1　通过相似用户结合评分 · · · · · · · · · · · · · 258
26.2　基于矩阵分解的模型 · · · · · · · · · · · · · · · 260
参考文献 · 263
索引 · 269

第 1 章　引　　言

人不应该过着野兽般的生活，而是要追寻美德与知识。

—— 但丁



1.1　学习与智能优化：燎原之火

优化是指为了找到更好的解决方案而进行的自动化搜寻过程。可以说，流程、方案、产品和服务之所以能持续改进，正是缘于优化为之**提供**的**强大动力**。优化不仅关乎方案的确定（从一些给定的可行方案中，选出最好的一个），**它还能主动创造出新的解决方案**。

优化催生了**自动化的创造和革新**。这看起来非常矛盾，因为自动化通常不会和创造与革新联系起来。因此，那些相信机器只能用来处理单调的重复性工作的人们在阅读本书时，会觉得书中的观点简直是胡言乱语，甚至会感受到如同被挑衅一般的愤怒。

自伽利略（1564—1642）之后，人们希望用科学改变世界，而这不仅需要哲学上的阐释，还需要**测量和实验**的支持。“测量那些可测量的，并使那些不可测量的变得可测量。”测量一开始看起来并不起眼，但它允许人们用务实的方式逐渐改变世界，只要人们还关心生产方式和生活质量。

几乎所有的商业问题都可以归结为**寻找一个最优决策值** $\boldsymbol{x}$， 这要通过**使某个收益函数** goodness($\boldsymbol{x}$) **最大化**来实现。为了能形象地理解，我们假设有一个集合变量 $\boldsymbol{x} = (x_1, \cdots, x_n)$，

它描述的可以是一个或多个待调节的旋钮，也可以是将要做出的选择，还可以是待确定的参数。在市场营销中，$\boldsymbol{x}$ 可以是一个向量，其数值表示为各类宣传活动（电视、报纸、各种网站、社交媒体）分配的预算，goodness($\boldsymbol{x}$) 则可以是由这些宣传活动而产生的新客户数量。在网站优化中，$\boldsymbol{x}$ 可以涉及图片、链接、话题和不同大小文本的使用，goodness($\boldsymbol{x}$) 则可以是该网站的普通访客成为客户的转化率。在工程学中，$\boldsymbol{x}$ 可以是一个汽车发动机的设计参数集，goodness($\boldsymbol{x}$) 则可以是该发动机每加仑汽油所能行驶的英里数。

将问题归结为“优化一个收益函数”也**激励着决策者，使用量化的目标，就可以用可衡量的方式来领会宗旨，也就可以专注于方针的制定而非执行的细枝末节**。当人们深陷于执行的泥潭中，以至于遗忘了目标时，企业就染上了“疫病”，此时如果外界环境发生了变化，这种“疫病”将会使企业无法做出及时的应对。

自动化是解决这个问题的关键：将一个问题形式化地表述后，我们把得到的收益模型输入计算机，计算机将自动创造出并找到一个或多个最佳的选项。另外，当条件和重点发生改变时，只需要修改一下收益函数的量化目标，再重启优化过程就可以了。当然，CPU 时间会是个问题，也并非每次都能保证找到全局最优解决方案。但可以肯定的是，使用计算机来搜寻，无论是速度还是范围，都远远领先于人力搜寻，并且这一领先优势会越来越明显。

然而，在大多数现实场景中，优化的惊人力量仍遭到很大程度的压制。优化在现实中没有被广泛采纳的主要原因是，**标准的数学优化理论假设存在一个需要最大化的收益函数**，也就是说，有一个明确定义的模型 goodness($\boldsymbol{x}$) 为每个输入配置 $\boldsymbol{x}$ 匹配一个结果。而目前，在现实的商业情境里，这个函数通常是不存在的。即使存在，靠人力找到这个函数也是极其困难、极其昂贵的。试想，问一个 CEO“请您告诉我，优化您业务的数学公式是什么”，显然不是咨询工作中开始对话的最佳方式。当然，一个经理对于目标应该会有一些想法和权衡，但是这些目标并没有以数学模型的方式给定，它们是动态的、模糊的，会随着时间改变，并且受限于估计误差和人们的学习进程。直觉被用来替代那些明确给定的、量化的和数据驱动的决策过程。

如果优化是燃料，那么点燃这些燃料的火柴就是**机器学习**。机器学习通过摒弃那种明确定义的目标 goodness($\boldsymbol{x}$) 来拯救优化：我们**可以通过丰富的数据来建立模型**。

机器学习与智能优化（learning and intelligent optimization，LION）结合了学习和优化，它从数据中学习，又将优化用于解决复杂的、动态的问题。LION 方法提高了自动化水平，并将数据与决策、行动直接联系起来。描述性分析和预测性分析之后，LION 的第三阶段（也是最终阶段）是**规范性分析**（prescriptive analysis）。**在自助服务的方式中，决策者手中直接握有更多的权力**，而不必求助于中间层的数据科学家。就像汽车的发动机一样，LION 包含一系列复杂的机制，但是用户（司机）并不需要知道发动机的内部工作原理，就可以享用它带来的巨大好处。在未来的几十年内，LION 方法带来的创新，将会像野火那样，以燎原之势延伸到大多数行业。那么企业就像野火频发的生态系统中的植物一样，只有适应并拥抱 LION 技术才能生存下来，并繁荣昌盛；否则，无论之前如何兴盛，在竞争逐渐加剧的挑战面前，都可能

土崩瓦解。

LION 范式关注的并不是数学上的收益模型，而是海量数据，以及如何针对多种具体选择（包括实际的成功案例）进行专家决策，或者如何交互地定义成功的标准。当然，这些都是建立在让人们感觉轻松愉快的基础之上的。例如，在市场营销中，相关数据可以描述之前的资金分配和宣传活动的成效；在工程学中，数据可以描述发动机设计的实验（真实的或模拟的）和相应的油耗测量方式。

1.2 寻找黄金和寻找伴侣

用于优化的机器学习需要数据。数据来源可以是**以往的优化过程**，也可以是**决策者的反馈**。

要了解这两种情境，先来看两个具体的例子。丹尼尔　克里金（Danie G. Krige，见图 1-1）是一名南非的采矿工程师，他曾遇到一个问题：如何在一张地图上找到挖掘金矿的最佳坐标 [74]。大约在 1951 年，他开创性地将统计学的思想应用于新金矿的估值，而这一方法仅需用到有限的几个矿坑。需要优化的函数是 Gold($\boldsymbol{x}$)，即坐标 $\boldsymbol{x}$ 处的金矿的金量。当然，在一个新的地方 $\boldsymbol{x}$ 评估 Gold($\boldsymbol{x}$) 是非常昂贵的。你可以想象，挖一个新矿没那么快，也没那么简单。但是在一些试探性的挖掘之后，工程师们会积累一些把坐标 $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3 \cdots$ 和金量 $\text{Gold}(\boldsymbol{x}_1), \text{Gold}(\boldsymbol{x}_2), \text{Gold}(\boldsymbol{x}_3)$ 关联起来的**实例知识**。克里金的直觉告诉他，用这些实例（**来自以往优化过程的数据**）可以建立起函数 Gold($\boldsymbol{x}$) 的模型。这个称为 GoldModel($\boldsymbol{x}$) 的模型归纳以往的实验结果，为地图上的每个位置 $\boldsymbol{x}$ 给出金量的估计值。通过优化，这个模型找到使预计黄金产量 GoldModel($\boldsymbol{x}$) 最大化的地点 $\boldsymbol{x}_{\text{best}}$，于是这个 $\boldsymbol{x}_{\text{best}}$ 成为下一个挖掘的地点。



图 1-1　丹尼尔　克里金，克里金法的发明者

可以用如图 1-2 所示的模型来形象地说明这个过程。先在地图上为每个矿坑插一根针，每根针的高度取决于在该处发现的金量。克里金的模型可以看作基于这些针的“训练”信息

在整个地图上方生成的一个曲面，使得给定位置的高度对应当地的预计黄金产量。因此，优化意味着在这个模型曲面上找到最高的那个点，并在对应的地点进行下一次挖掘。

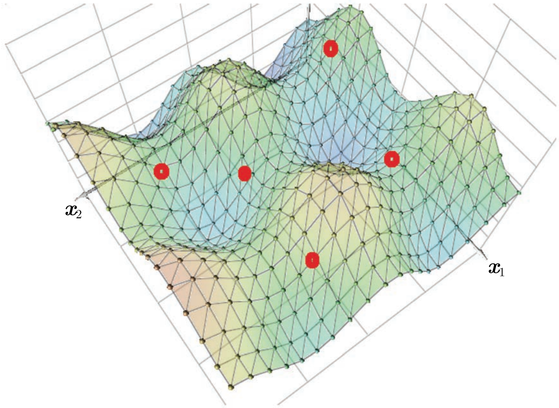


图 1-2　从样本中使用克里金法构造模型。一些样本在图中用点标示出来。表面的高度和颜色依赖于产金量（另见彩插）

这种技术现在被称为克里金法（Kriging），它背后的理念是未知点对应的值应该是其邻近已知点所对应的值的加权平均，权重与这些已知点到该未知点的距离相关。高斯过程、贝叶斯推断和样条函数（spline）都涉及了相关的建模方法。

第二个例子关于**决策者的反馈**。想象有这样一个约会服务：人们付费在数以百万计的候选人中匹配一个最佳的约会对象。在克里金法中，需要优化的函数是存在的，只是评估起来极为困难。对于这个案例，我们很难假设存在一个类似的函数 IdealMate($\boldsymbol{x}$)，它将个人特征 $\boldsymbol{x}$，例如美貌、智力等，与你的个人喜好联系起来。如果你不这么认为，且坚信存在这样一个函数，那么给你留一个作业，尝试用准确的数学术语来定义你心目中理想伴侣的 IdealMate 函数。即使你能准确地指出某些组成部分，例如 Beauty($\boldsymbol{x}$) 和 Intelligence($\boldsymbol{x}$)，但是在开始寻找最佳候选人之前，把这两个目标合并起来仍然是困难的。像“降低多少 IQ 值对应减少一点美貌”或者“美貌是否比智力重要，重要多少”这类问题是非常难回答的。假使你很痛苦地给出了一个初步答案，也肯定不会相信这个优化，在真正见到这个候选人之前，你不会为这个匹配服务付费，当然也不会对服务感到满意。你会想了解这个人的特征，而不仅仅是得到系统优化的肤浅的 IdealMate($\boldsymbol{x}$) 函数值。只有在考虑过不同的候选人并且对这个匹配服务进行反馈后，你才能希望找到最满意的另一半。

换句话说，在一开始，待优化函数中的某些信息是不全面的，只有决策者才能够调整优化的过程。**许多现实问题，即使不是大多数，都需要借助有学习参与的迭代过程来解决**。在了解了越来越多的案例后，用户会认识并调节自己的喜好，系统会从用户的反馈中建立起他的喜好模型。这一过程将持续下去，直到用户满意或者直到耗尽为这一决策分配的时间。

1.3 需要的只是数据

下面继续谈论商业用户的动机。如果你不关心这方面的内容，可以放心地跳过这部分，直接阅读 1.6 节。

商业领域里充斥着各种数字形式的数据。大数据指的是大量的半结构数据。顺便提一句，在 20 世纪七八十年代，数据对于当时的存储设备来说是庞大的，而如今的“大数据”更多是商业上的宣传概念：即便是最大的公司产生的所有数据，只需一台 PC 就足以处理了。

随着社交网络的爆发、电子商务的迅速扩张和物联网的兴起，网络正在掀起一场由结构化和非结构化数据引起的海啸。这场海啸驱使人们在信息技术领域花费多达数十亿美元。也有新的证据表明，标准的商业智能平台使用率正在下降，这是因为企业界已经不得不开始考虑一些非结构化的数据，而这些数据拥有无法估量的现实价值。例如，社交网络产生大量的数据，其中的大多数无法分类，也无法用传统数据的刚性层次结构来表示。试想，你该如何评估 Facebook 上一个“赞”的价值？况且非结构化数据需要用自适应方法来分析。再想想，随着时间的流逝，一个“赞”的价值会发生怎样的变化？由于这类问题的存在，我们需要在数据建模、自适应学习和优化等领域运用更加先进的技术。

为了让软件能够自我改进，并能快速适应新数据和调整后的业务目标，需要使用 LION 方法。这种方法的优势在于能够从过往的经验中学习、**在工作中学习**、应对不完全的信息，并快速适应新的情况，而这些能力通常只与人类的大脑联系起来。

LION 技术这种内在的灵活性是至关重要的，因为在求解过程开始之前，我们很可能无法确定哪些是对决策有影响的因素和重点。例如，我们要给一个市场营销的前景评分来估计其价值，应该考虑哪些因素？这些因素又对结果分别有多大程度的影响？如果使用 LION 方法的话，这些问题的答案就是：“这些都不是问题。”系统会开始自我训练，源源不断的数据加上终端用户的反馈将快速提升系统的性能。专家——这里指营销经理——可以通过表达他们自己的观点来改善系统的输出。

1.4 超越传统的商业智能

每一家企业都需要数据来满足 3 项基本需求：

(1) **了解**目前的业务流程，并评估以往的表现；

(2) **预测**商业决策的影响；

(3) 对业务的关键因素制定并执行明智且合理的决定，从而**提升**赢利能力。

传统的**描述型**商业智能（business intelligence，BI）擅于记录和可视化过往的表现。构建这样的记录意味着需要聘请顶级顾问，或雇用那些有统计、分析和数据库等领域知识的专业人员。专家必须要设计数据提取和操作的流程，然后交给程序员来实际执行。这是一个缓慢而繁琐的过程，毕竟大多数商业的境况都是瞬息万变的。

因此，那些严重依赖于 BI 的企业正在利用性能快照，尝试理解当前情况和未来趋势，并对此做出反应。这就如同开车的时候只盯着后视镜，很有可能会撞上什么东西。现在对于企业来说，就像是已经撞到了一堵僵化的墙，并且缺乏快速适应变化的能力。

预测分析确实在预见方案效果方面做得更出色，然而，**将数据驱动模型和优化进行整合，**自动创建完善的解决方案，才是 LION 真正的强大之处。**规范性分析**做到了引领我们直接从数据到最佳改进方案，**以及从数据到可执行的洞察力，再到行动本身!**

1.5 LION 方法的实施

对于处在不同业务状态的企业而言，全面采用 LION 方法作为商业实践的步骤会有所不同。更重要的是，相关数据的情况也会影响这一进程。显然，在数据收集完成的时候引进 LION 范式会相对容易，开销也更少。对某些企业来说，由于遗留系统的迁移和转换需要涉及大范围的整理，开销会非常大。这也正是那些老练的服务提供商能大显身手的地方。

除了整理和定义相关数据的结构之外，最重要的一点就是建立起数据分析团队和商业终端用户之间的合作。LION 方法通过自身的特性提供了一种合作方式，助其共同发现蕴藏在结构化或半结构化数据中的潜能。数据分析团队能够和商业终端用户高效地并肩合作，关键在于能够使业务目标的不断变化迅速反映到模型上。LION 方法的引入可以帮助数据分析团队在价值创造链中产生根本性的变化，它能揭示隐藏的商机，也能加快他们的商业伙伴对客户要求和市场变化的反应速度。

就业市场也将被打乱。从人类的实例中进行学习的软件将推导出我们在使用却又不明确了解的规则。这将消除进一步自动化的障碍，在许多需要适应性、常识和创造性的任务中，机器将会代替工人，也许会让中产阶级处在风险之中 [110]。

LION 方法可以说是一种极具颠覆性的**发现隐藏价值的智能方法，它能快速适应改变并改进业务**。通过恰当的规划和实施，LION 技术可以帮助企业在竞争中独领风骚，避免被燎原之火灼伤，同时也可以帮助个人在高技能人才的就业市场中保持竞争力。

1.6 “动手”的方法



因为这是一本关于从实例中进行（机器）学习的书，所以在学习这本书时也要遵从这一点。本书大多数的内容都是按照从实例中学习和**从实践中学习**的原则来安排的。当介绍不同的技术时，我们会讨论这些技术的基础理论，然后会总结出一些你“应该了解的梗概”。本书鼓励用现实中的情况来做实验，你可以在本书的网站上找到相关的例子和软件。这样做能让你体会到 LION 技术并不是只为专家准备的；它属于任何对快速且可测量的结果感兴趣的实践者。

第一次阅读本书时你可以跳过某些理论部分。但是某些理论知识是十分关键的，它们不仅能帮助开发新的、更加先进的 LION 技术，还能使你更加得心应手地使用这些技术。掌握一些基础的、未被稀释的理论，就像在陌生国度旅行时手中有地图指引。如果你是一艘不知要驶向何处的船，那么风往哪边吹都是无意义的。

我们会尽量兼顾开发人员和终端用户的感受。下面两个图标粗略地表示了不同章节的难度级别。当然，难易程度的真实感受跟读者的知识背景有关，因此可能与我们试验性的级别分类不同。



容易的话题



进阶的话题

本书作者以及读者群发布的数据、指导说明和教学短片都可以在本书的网站上找到：https://intelligent-optimization.org/LIONbook/。

我们感谢为这本书做出了贡献的人们。首先是照片和插画。Carlo Nicolini 提供了在 LION-4@VENICE 2010 会议期间拍摄的威尼斯照片。第 1 章的但丁像是 Domenico di Michelino 于 1465 年在佛罗伦萨完成的。George Chernilevsky 提供了第 2 章装着蘑菇的篮子的图片。第 9 章大脑图片是达 芬奇（1452—1519）的作品。聚类深度网络的图来自 Geoffrey Hinton。第 11 章的 Vapnik 教授的照片由 Yann LeCun 提供。超限学习机的图片来自 Guangbin Huang。储备池的结构图由 Herbert Jaeger 提供。第 13 章的威尼斯绘画由卡纳莱托在 1730 年完成。第 15 章的绘画是米开朗基罗于 1541 年完成的。我们也在维基百科中找到了一些解释性的图片。Hopfield 网络图来自 Gorayni，能级相图由 Mrazvan22 提供。本书作者和作者的儿子们都是维基百科条目积极的撰写者。第 14 章章首 Reschense 湖的照片来自 Markus Bernet。第 10 章的蟾蜍图片由 André Karwath 提供。

最后，我们感谢读者为提升这本书的品质所做的越来越多的贡献。他们包括 Patrizia Nardon、Fred Glover、Alberto Todeschini、Yaser Abu-Mostafa、Marco Dallariva、Enrico Sartori、Danilo Tomasoni、Nick Maravich、Drake Pruitt、Dinara Mukhlisullina、Rohit Jain、Jon Lehto、George Hart、Markus Dreyer、Yuyi Wang 和 Gianluca Bortoli。书中的漫画是 Marco

Dianti 赠予我们的礼物。我们十分乐意与读者沟通。如果你有评论、建议或者勘误[1]，请给我们发电子邮件，我们会把你的名字加在下一个版本中。你可以在 LIONlab 的网站上找到联系方式和电子邮件地址：https://intelligent-optimization.org/。

第 2 版补遗

现在你正在读的是本书的第 2 版：我们在此感谢许多读者发送的更正和改进建议。

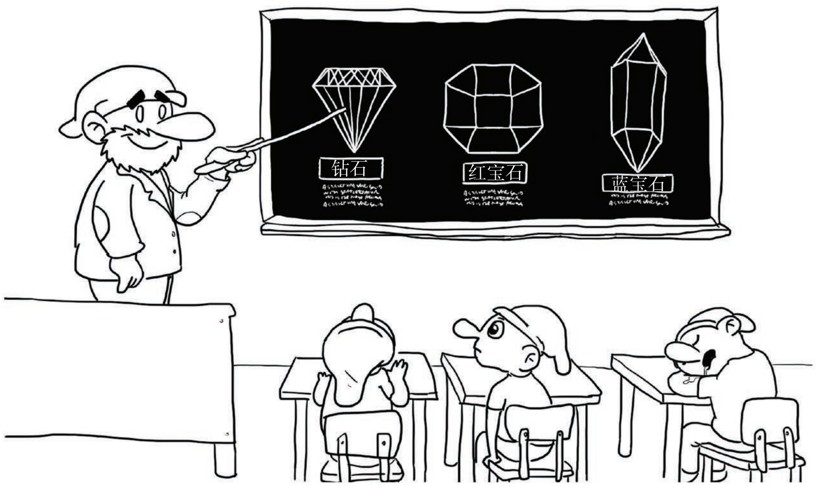
电子书

扫描如下二维码，即可购买本书电子版。

① 中文版勘误请读者到图灵社区的本书页面提交：http://www.ituring.com.cn/book/1413。—— 编者注

第 2 章　懒惰学习：最近邻方法

自然不允许跳跃。



如果你还记得小时候是如何识字的，那么你就可以理解什么是**从实例中学习**，尤其是监督学习。父母和老师给你展示一些带有英文字母（a、b、c，等等）的实例，然后告诉你：这是 a，这是 b，

当然，他们并没有用数学公式或者精确的规律来描述这些字母的几何形状。他们只是展示了一些不同风格、不同形式、不同大小和不同颜色的**已标记的实例**。经过一些努力和失误之后，你的大脑就能够正确识别这些实例了。然而这不是关键，因为仅凭记忆你其实就能够做到这一点。重要的是，通过这些实例的训练，你的大脑还能从中提取出与认字真正相关的模式和规律，过滤掉不相关的“噪声”（比如颜色），从而进行**泛化**（generalize），以**识别**在训练阶段从未见过的**新实例**。这是很自然的结果，但确实是值得注意的成果。取得这一成果不需要什么先进的理论，也不需要博士学位。如果有一种方法也能如此自然而又轻松地解决商业问题，是不是很令人振奋呢？结合了从数据中学习和优化的 LION 范式就是这样的一种方法，我们将从这一熟悉的语境开始。

在**监督学习**中，由监督者（老师）给出一些**已标记的实例**，系统根据这些已标记的实例来完成训练。每一个实例是一个数列，它包括一个作为输入参数的向量 $\boldsymbol{x}$，称为**特征**（feature），和与之相对应的输出标记 y。

本书作者生活的地方有很多的山地和森林，因此采蘑菇是一项十分普及的消遣活动。虽然采蘑菇很受欢迎也很有趣，但是误食有毒的蘑菇将造成致命的危害（见图 2-1）。这里的小孩子在很小的时候就学会了如何区分可以食用的和有毒的蘑菇。到这里来的游客可以买到相关的书籍，书中有这两类蘑菇的图片和特征；他们也可以把采到的蘑菇带到当地的警察局，让专家帮他们免费检验这些蘑菇。



图 2-1　采蘑菇要区分可以食用的和有毒的

这里有一个被简化过的例子，如图 2-2 所示，假设我们用两个参数，比如高度和宽度，就能够区分这两种蘑菇。当然，一般来说，我们需要考虑更多的输入参数，像颜色、形状、气味等，甚至是更加令人困惑的正类（可以食用的）和负类实例的概率分布。

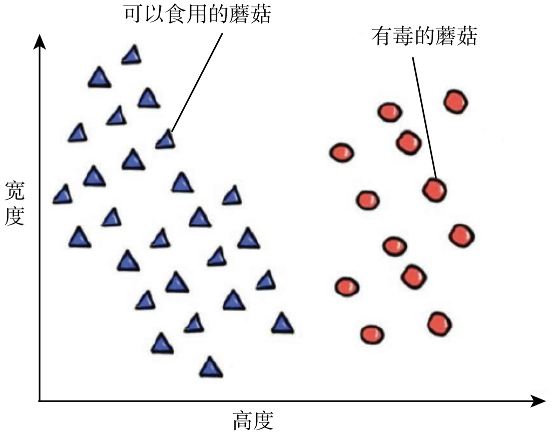


图 2-2　简化的例子：两个特征（宽度和高度）用以区分可以食用的和有毒的蘑菇

那些懒惰的初学者在采蘑菇的时候遵循简单的模式。通常他们在采摘蘑菇之前没有学习任何相关的知识，毕竟，他们到特伦蒂诺是来度假的，而不是来工作的。当发现一个蘑菇时，

他们会在书中寻找相似的图片，然后仔细检查对照细节列表中的相似特征。这就是**机器学习中懒惰的“最近邻”**（nearest neighbor）**算法**在实际问题中的一次应用。

为什么这样一种简单的方法是有效的呢？我们可以用 Natura non facit saltus（“自然不允许跳跃”的拉丁文）原则来解释它。自然的事物与特征常常是逐渐改变，而不是突然改变的。如果你将书中的一个可食用的蘑菇作为原型，然后发现你自己采摘的蘑菇与这个原型蘑菇的各项特征非常相似，那么你也许会认为你的蘑菇是可以食用的。

声明：不要使用这个简单的例子来区分真正的蘑菇，因为每一种分类器都有一定的概率出错，另外蘑菇分类中的*假正类*（把有毒的蘑菇当成可食用的）将会对你的健康造成极大的损害。

最近邻方法

在机器学习领域，**最近邻方法**的基本形式与**基于实例的**学习、**基于案例的**学习和**基于记忆的**学习有关。它的工作原理如下：我们把已标记的实例（包括输入及相应的输出的标记）储存起来，不进行任何操作，直到一个新输入模式需要一个输出。这种系统被称为**懒惰的学习者**：它们只是将这些实例储存起来，其他的什么也不做，直到用户询问它们。当一个新输入模式到达时，我们在存储器中查找到与这个新模式相近的那些实例，输出则由这些相近模式的输出所决定，见图 2-3。一百多年来，这种数据挖掘的形式仍然被统计学家和机器学习专家广泛地用于分类问题和回归问题。

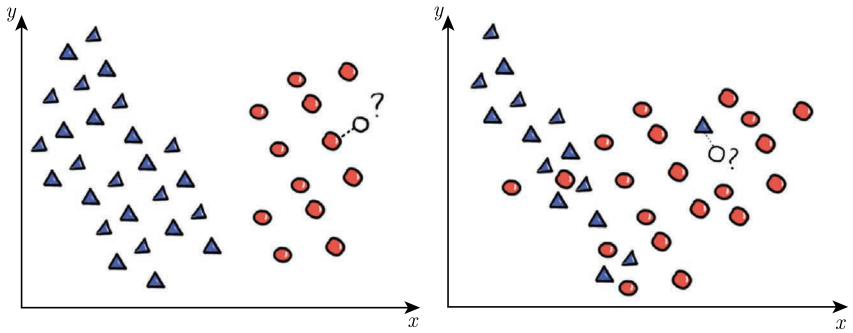


图 2-3　最近邻分类：一个清晰的情形（左），一个不太清晰的情形（右）。在第二种情形中，标有问号的查询点的最近的邻居属于负类，但它的更多近邻属于正类

简单点说，一个新输入对应的输出就是存储器里相距最近的那个实例的输出。如果要判断一个新遇见的蘑菇是否可以食用，我们就把它归到记忆中与之最相似的蘑菇的那一类。

虽然十分简单，但很多情况下这种技术都出奇地有效。然而它毕竟是一种偷懒的方法，要为懒惰付出代价！不幸的是，为识别一个新实例所花费的时间可能与存储器中的实例数量成

正比，除非用不那么偷懒的方法。这就好比有一个学生，虽然平常买了不少书，但是只在遇到问题时才去读这些书。

一个更具健壮性和灵活性的方法是考虑大小为 k 的近邻集合，而不仅仅是最相近的那一个，不难猜到这种方法被称为 **K 近邻**（KNN）方法。它的灵活性来源于可以使用不同的分类方法。例如，新实例的输出可以用**多数同意规则**，即输出这 k 个近邻中占大多数的那一个输出。如果想要更加安全的方法，可以仅在这 k 个近邻的输出完全相同时才确定新实例的类别（**一致同意规则**），否则就输出"未知"。这个建议可以用在区分有毒的蘑菇时：如果输出"未知"，就联系当地警方寻求帮助。

如果面临的是一个**回归问题**（预测一个实数，例如蘑菇中有毒物质的含量），我们可以将这 k 个最相近的实例的输出平均值作为新实例的输出。

当然，这 k 个实例到新实例的距离可能有所差别，而且在某些情况下，距离较近的实例对新实例的输出影响更大是很合理的。在这种被称为**加权 K 近邻**（WKNN）的方法中，权重取决于距离。

设给定的正整数 $k \leqslant \ell$（ℓ 为已标记实例的个数）；$\boldsymbol{x}$ 表示新的实例，是一个属性向量[①]。下面是一个用于估计 $\boldsymbol{x}$ 所对应的输出 y 的简单算法，分两个步骤。

(1) 在训练集中找到 k 个下标 $i_1, \cdots, i_k$，使得属性向量 $\boldsymbol{x}_{i_1}, \cdots, \boldsymbol{x}_{i_k}$ 与给定的 $\boldsymbol{x}$ 最相近（根据某种给定的属性空间度量）。

(2) 通过下面的加权平均来计算估计的输出，权重反比于属性向量之间的距离：

$$y = \frac{\displaystyle\sum_{j=1}^{k} \frac{y_{i_j}}{d(\boldsymbol{x}_{i_j}, \boldsymbol{x}) + d_0}}{\displaystyle\sum_{j=1}^{k} \frac{1}{d(\boldsymbol{x}_{i_j}, \boldsymbol{x}) + d_0}} \tag{2.1}$$

其中 $d(\boldsymbol{x}_i, \boldsymbol{x})$ 指两个向量在属性空间中的距离（例如欧氏距离），d_0 是一个小的偏移常数，用以避免出现 0 作为除数的情况。d_0 越大，距离较远的点的贡献就越大。如果 d_0 趋近于无穷大，那么这 k 个实例的权重就几乎一样了。

WKNN 算法很容易实现，并且相应的估计误差也很小。它的主要缺点是需要大量的内存空间，以及在测试阶段巨大的计算量。因此我们常常将已标记的实例进行聚类，用来减少所需的内存空间。聚类方法按照相似性将它们划分成一个个小组，并且只存储每个小组的原型（中心）。第 15 章会讨论更多的细节。

本书接下来将继续考虑新实例和内存中实例之间的距离，并且将这一想法一般化。**核方法**与**局部加权回归**就可看作最近邻方法的一般化，这两种方法并不是粗鲁地直接将远处的点排除，而是根据它们到查询点的距离，灵活地赋予它们相应的重要性（权重）。

① feature vector 若译为"特征向量"，恐被误认为 eigen vector，因此译作"属性向量"。——译者注



梗　概

KNN（K 近邻）是一种原始的懒惰的机器学习方式：它只是把所有的训练实例存在存储器中（输入和对应的输出标记）。

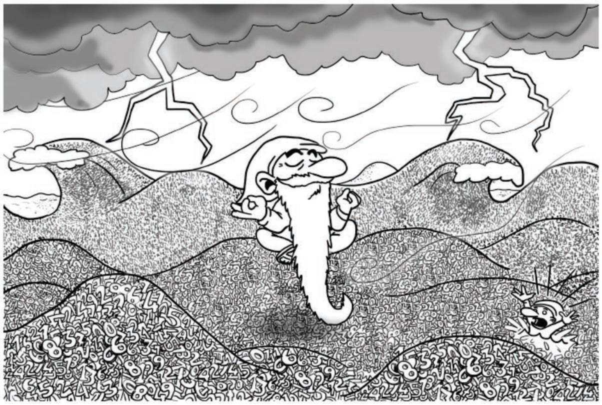
当有一个新输入并需要计算其对应的输出时，在存储器中查找 k 个最接近的实例。读取它们的输出，并根据它们的大多数或平均值推导出新实例的输出。当存储了非常多的实例时，训练阶段的懒惰会让预测阶段的响应时间变得很长。

相似的输入经常对应着相似的输出，这是机器学习领域的一个基本假设，因此 KNN 方法在很多实际案例中都有效。它与人类的某些“基于案例”的推理具有相似性。虽然这个方法简单粗暴，但它在很多现实案例中的效果都令人惊奇。

从现在起，不要做一个懒惰的学习者，别以为这样可以高枕无忧。继续读下面的章节，坚持学下去。早起的鸟儿有虫吃，睡懒觉只能肚子空空了。

第3章　学习需要方法

数据挖掘，名词，对数据进行的严刑逼供
如果拷打得足够久，它会向你坦白任何事情。



无论是对于人类，还是对于机器来说，学习都是一种强大却又微妙的能力。真正的学习涉及如何从一个现象中提取深层次的、基础的关系，如何简要地概括各种不同的事件所遵循的规律，以及**如何通过发现基本的定律来统一解释不同的情况**。

最重要的是，我们真正的目标是**能够泛化的模型，以及模型对新实例的解释能力**，新实例是指与训练实例来自同一个应用领域，但在学习阶段没有遇见过的实例，而从实例中学习仅仅是走向这一终点的途径之一。与此相反，死记硬背常常被认为是非常低效的学习方式，它虽然对初学者有一定的作用，但是无法使你成为真正的专家。如果目标是泛化，那么模型在学习集上的表现并不能保证泛化是正确的，还可能导致我们对结果过于乐观，因此要**极其谨慎地估计这个模型的性能**。归根结底，只擅于死记硬背的学生日后在生活中未必能取得个人的成功。

我们需要定义机器学习（简称为 ML）的上下文，让其能够全力发挥，又不会因使用不当或过于乐观而造成损害。另外，使用 ML 并不意味着就不需要使用我们的头脑了。

事实上，开始机器学习流程之前，用户会根据直觉和智能在原始数据中提取一个具有代表性的子集，这一步是非常有用的。特征（或属性）是观察到的现象的各个可度量的性质，这些性质包含了与输出有关的有用的信息。这一准备阶段称为**特征选择**（选出一个集），以及**特**

征提取（生成一个组合，见图 3-1）。

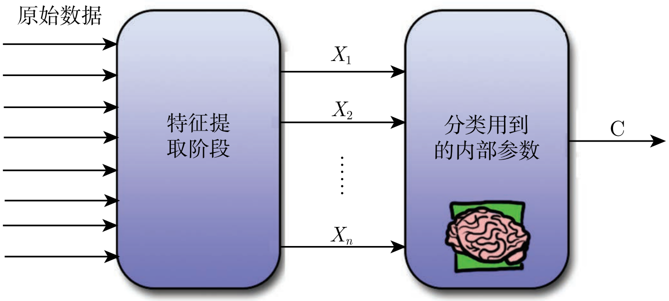


图 3-1　监督的学习体系：特征提取和分类

举例来说，字母和数字的图像可以作为输入，输出则是与图像对应的字母或数字符号。与此相关的应用包括邮政编码的自动读取、旧书页图片自动转换为相应文本内容等，这些被称为光学字符识别。直觉告诉我们，图片的绝对亮度不是一个能提供信息的属性（无论亮度如何，数字都保持不变）。在这种情况下，合适的属性可能与图像中的边缘或灰度直方图等有关。一些更为复杂的技术尝试确保那些经过平移和放大的图像也能被正确识别，例如在提取特征的时候参考图像的重心（将一个像素的灰度值当作那个点的质量），或者将图像进行伸缩，使得黑色部分面积尽可能大，等等。提取有用的特征通常需要对该问题的专注、见解和知识储备，这样做将大大简化接下来的自动学习阶段。这就好比一位学识渊博的教授为他所擅长的一门课精心准备教材。

考虑 ℓ 元组（元素的有序列表）的训练集，其中每一个元组是形如 $(\boldsymbol{x}_i, y_i), i = 1, \cdots, \ell$ 的有序对，$\boldsymbol{x}_i$ 是一个 d 维空间里（$\boldsymbol{x}_i \in \mathbb{R}^d$）的输入参数向量（数列），$y_i$ 是测量到的输出，即算法要学习的部分。如前所述，我们将会考虑两类问题：当 y_i 可以取实数值时，为*回归问题*（regression）；当 y_i 在一个有限集里取值时，为*分类问题*（classification）。

分类问题（识别以特征 $\boldsymbol{x}$ 描述的某一特定目标的类别）中，输出是类别的相应编码。输出 y 属于一个有限集，例如 $y_i = \pm 1$，或者 $y_i \in \{1, \cdots, N\}$。例如，可以将蘑菇分为两类：可食用的和有毒的。

回归问题的输出从一开始就是一个实数值，它的目标是通过建模研究因变量（输出值 y）与一个或多个自变量（输入值 $\boldsymbol{x}$）之间的关系。例如，根据蘑菇的特征来预测其有毒物质的含量。

在某些情况下，分类很难始终保持清晰，因为不同类别之间的界线可能是模糊的。试想如何区分秃顶的人和有头发的人？二者之间并不存在明确的界线，正如为掉头发而焦虑的人和卖防脱发产品的人所知的那样。

这种情况下，“清晰的”分类问题很自然地转换成回归问题。为了谨慎起见，输出可以是 0~1 的实数值。对于给定的输入值，我们可以认为这个输出值是**某个给定类别的后验概率**；如

果不能以概率来解释的话，也可以当作**模糊隶属度**。举例来说，如果有个人只有几根头发了，说他有头发的概率是 0.2 并没什么意义，这种情况下说他以 0.2 这个值模糊隶属于有头发的人可能更合适。当实验的输出数据不确定但可重复时，使用概率是合适的。

以连续值作为输出，例如 0~1，可以增加分类系统在实际使用中的**灵活性**。还可以通过设定阈值，判断是由人还是由更复杂的系统来帮助解决一些令人迷惑的案例（例如案例的输出落在 0.4~0.6 的范围内）。简单明了的案例由系统自动处理，而最棘手的案例则由人来处理。在光学字符识别中，比如有一个图像，它可能是数字 0（零），也可能是字母 O（就像单词 Old 中的那个），系统最好告诉我们每种情况都有 50% 的可能性，而不是强行做一个硬分类。当然，接下来还可以使用这个字符的邻近字符或语义信息来进一步分辨。

3.1　从已标记的案例中学习：最小化和泛化

监督学习方法使用实例构造一个函数 $y=\hat{f}(\boldsymbol{x})$，将输入 $\boldsymbol{x}$ 和输出 y 关联起来。这一关联选自一个**灵活的模型** $\hat{f}(\boldsymbol{x};\boldsymbol{w})$，其中的灵活性来自**可调整的参数**（即**权重系数**）$\boldsymbol{w}$。

为了能有具体的印象，想象一台将输入转化为输出的绞肉机，可以通过齿轮与杠杆来调节它。或者想象一个等待输入的“多功能盒子”，它能根据内部参数的影响产生输出。用于“自定义”这个盒子的信息取自给定的训练实例集。ML 的神奇之处在于，这些齿轮的调节不是手动完成的，而是自动通过正确的输入–输出对的示例来进行优化。

图 3-1 展示了该架构的一种方案，其中区别了两个部分，即特征提取，以及分类器内部最优权重的确定。在许多情况下，特征提取需要一些来自人类的洞见，然而**最优参数的确定则是完全自动的**，这也是这一方法被称为机器学习的原因。**让模型对**训练集**中的实例进行正确的分析**，从而确定那些自由参数。

认为优化具有强大力量的真正信徒会先从定义*误差度量*（error measure）最小化开始[①]，并通过合适的（自动化的）优化过程来确定最优参数。这里的*误差度量*指所有正确答案（由实例的标记得出）与模型输出（由这个多功能盒子的输出得出）之间误差的总和。通常这个误差是一个绝对值，并经常取其平方值。**“误差平方和”**可能是机器学习领域应用最为广泛的一种误差度量。如果误差为零，表明这一模型在给定的实例上能百分之百地正确工作。误差越小，模型在这些实例上的平均表现就越好。

监督学习因此变成了**最小化某个特定的误差函数**，这一误差函数依赖于参数 $\boldsymbol{w}$。如果只关心最终结果，你可以把优化部分当作多功能盒子上的**红色大按钮**，当你将已标记的实例集输入这个盒子后，按下这个按钮，它会就某个具体问题提供自定义的结果。

如果你有兴趣开发新的 LION 工具，接下来的章节将为你呈现优化技术的更多细节。它们的要领是，如果函数是光滑的（想象一下宜人的草木丰茂的加州群山），人们可以蒙住眼睛，

① 把要优化的函数乘以 (-1)，就可以将最小变为最大。这就是为什么经常谈论“最优化”时通常不会谈及特定方向的最大或最小的原因。

像跳伞那样找到一个随机初始点，然后用脚来感觉周围的地势，并总是向着最速下降的方向移动，这样就可以找到海拔很低的那些点（如湖泊）。计算机并未配备“人类视觉”来“看到”这些湖泊，只能每次抽样一个点。通过重复两个步骤，在当前点的邻域进行抽样——在 $\boldsymbol{w}$ 空间中——并移动到误差更小的邻居，可以生成一个值越来越小的轨迹。神奇的是，对许多应用而言，这个简单的过程足以达到适当的 $\boldsymbol{w}^*$ 值。

现在用数学的语言来表述。如果需要优化的函数是可微的，一个简单的方法是使用**梯度下降**（gradient descent）。人们可以重复地计算这个函数关于权重的梯度，并朝着负梯度的方向移动一小步。事实上，这是神经网络里很流行的一种技术，称为基于误差**反向传播**（backpropagation）的学习[115, 116, 92]。

函数是平滑的，这一假设并非凭空捏造，而是基于监督学习的一个基础的**平滑假设**：若两个输入 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 距离很近[①]，则它们对应的输出 y_1 和 y_2 应该也很相近。如果这一假设不成立，那么就不可能将有限训练集泛化到可能无限多的尚未见过的新测试案例上。可以注意到，人们大脑中的信号和相互作用的物理现实都满足这一平滑假设。树突中的化学和电信号交互可看作神经元的输入，神经元的活动（输出）平滑地依赖这些输入，并依次作出反应。

到现在，你可能会认为机器学习等同于对训练集上的某性能度量的优化，但还缺失了一个部件。误差函数的最小化是首要的关键因素，但并不是唯一的。如果**模型复杂度**（灵活性、需要调整的参数个数）极高，模型在训练实例上达到零误差是非常容易的，但若将这个模型用于预测新实例的输出，则很可能会一塌糊涂。用人类的学习来打比方，如果死记硬背却未抓住真正的规律，学生将很难做到举一反三。这与*偏差–方差*（bias-variance）困境有关，它要求我们在选择模型的时候倍加小心，或者将目标函数最小化，改为训练实例误差与模型复杂度的加权组合。

偏差–方差困境可表述如下。

- 参数过少的模型会因较大的偏差而失准：它们缺乏灵活性。
- 参数过多的模型则会因较大的方差而失准：它们对于样本中的细节过于敏感（细节中的改变将会使模型产生大的变化）。
- 找到最佳模型需要控制“模型复杂度”，即模型的结构和参数数量都要恰到好处，从而在偏差和方差之间达成折中方案。

避免过于复杂的模型，而优先选用简单的模型，这种偏好有一个有趣的名字：*奥卡姆剃刀*，意指“剃去”理论中那些不必要的复杂性[②]。优化仍在使用，但由于顾及模型的复杂度问题，误差度量需要进行整合。

---

① 有些情况下可以测量 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 之间的标准欧几里得距离，但其他情况下需要更有针对性的度量方法。

② 奥卡姆剃刀归功于 14 世纪的神学家和方济会修士奥卡姆的威廉，他写道：“如无必要，勿增实体（entia non sunt multiplicanda praeter necessitatem）。”引用艾萨克 牛顿的解释：“我们要承认，无须为自然事物寻找更多的原因，能正确并充分地解释事物的表现就够了。因此，对于相同的自然现象，我们尽可能给出相同的原因。”

区分监督分类的两类方法也是有意义的。第一类热衷于得到某个关于输入是如何产生输出的“构造性的模型”；第二类更在意结果，即获得正确的分类。前者关心对内在机制的解释，后者则单纯地在意其性能。

第一类情况下，**生成方法**（generative method）尝试在实例中建模，为不同的类型 y 生成实测数据 $\boldsymbol{x}$ 的过程进行建模。给定某个类，比如有毒的蘑菇，它具有某种外形的概率是多少？用数学的术语来说，学习到的是一个类条件概率密度 $p(\boldsymbol{x}|y)$，即在给定 y 的情况下 $\boldsymbol{x}$ 的概率。那么，在给定一个新测量 $\boldsymbol{x}$ 时，根据贝叶斯定理，分类 y 可以通过最大化后验概率得到：

$$p(y|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y)p(y)}{\sum_y p(\boldsymbol{x}|y)p(y)} \tag{3.1}$$

其中 $p(\boldsymbol{x}|y)$ 称为数据的似然性，$p(y)$ 是先验概率，用以反映在测量之前各种结果的可能性。分母中的项就是普通的规范化条件，使得概率之和为 1。有个用于帮助记忆贝叶斯定理的口诀：后验 = 先验×似然性。

判别算法（discriminative algorithm）就不会尝试建模数据的生成过程，它们直接估计 $p(y|\boldsymbol{x})$，这个问题在某些情况下比之前生成方法的两步过程（首先建模 $p(\boldsymbol{x}|y)$，然后才导出 $p(y|\boldsymbol{x})$）要更简单。判别型方法的例子包括多层感知器神经网络，以及支持向量机（SVM）等，接下来的章节里将会讨论。

判别算法所示的捷径具有深远意义，我们**不必知道某些类别如何产生输入实例，也不必为此建立一个详尽的模型，就可以构造精确的分类器**。想要不用冒着生命危险去采摘蘑菇，并不需要成为真菌学家，你只需要大量有代表性的蘑菇实例集，并且它们已经正确地分好了类。

认识到不需要成为某个领域的专家就可以做出贡献，这是个人的一小步，却是 LION 发展道路上的一大步。不用说，成功的企业用朴实低调而又功能强大的数据驱动和优化驱动的工具，弥补了专业知识方面的缺憾。

3.2　学习、验证、测试

基于已标记实例的学习要求我们采用**细致的实验程序**来测量学习过程的效果。尤其注意，不能将已经用于训练的实例再用于测试学习系统的性能，如果这么做，将是一个可耻且无法原谅的错误。机器学习的目标是获得一个拥有**泛化**能力的系统，用以分析新的或以往未见过的数据；否则，这个系统就不是在学习，而只是记住了一些已经知道的模式，这也是学校不停更换考试题的原因

假设有一个能从给定的概率分布中生成标记实例的监督者（一个软件程序或实验过程）。在训练阶段，我们最好向监督者索要一些实例，在测试性能阶段再索要一些新的实例。理想情况下，用于训练的实例数量应足以确保收敛，并且用于测试的实例数量也应该足以保证这个估计具有统计学意义。如果一个用于区分可食用蘑菇和毒蘑菇的机器学习系统只测试了 7 个蘑菇，那么我们强烈建议你不要认为这个系统是可用的。

然而现实可能与理想相差甚远。一些情况下，训练集是相当小的，并且需要尽可能保证它们能同时满足训练和性能测试的要求。这种情况下，实例集必须清楚地分为**训练集**和**验证集**，前者用来训练，后者用来测试性能，如图 3-2 所示。一个典型的性能测试是系统输出与监督者给出的正确输出之间的**均方根**（root mean square，RMS）误差。值集的 RMS 值是原始值的平方的算术平均的平方根。若 e_i 是第 i 个实例的误差，则 RMS 值由下式给出：

$$\text{RMS} = \sqrt{\frac{e_1^2 + e_2^2 + \cdots + e_\ell^2}{\ell}}$$

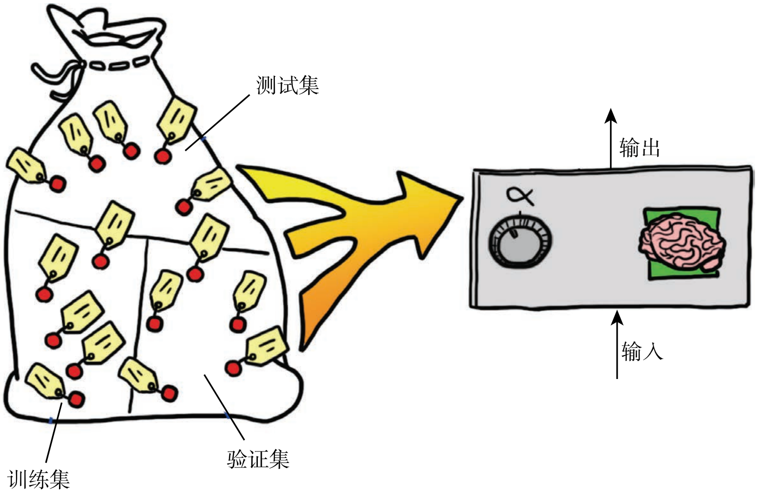


图 3-2 标记的实例必须分为训练集、验证集和测试集

一般而言，学习过程通过优化模型参数以使得模型尽可能好地拟合训练数据的输出。那样的话，如果我们从验证数据的同一个总体中取一个独立的抽样作为训练数据，一般会导致验证数据集的误差大于训练数据集的误差。如果训练过度的话，这种差异很可能会变得非常严重，并导致**过拟合**（**过度训练**）。当训练实例很少，或者模型中的参数很多时，更容易发生这种情况。

如果实例数量非常有限，就会面临一个问题：我们是希望用其中的大多数来训练，但要承担一个差劲的有噪声的性能测量的风险，还是拥有一个更具健壮性的性能测量，但要放弃一些训练实例？具体来说，如果有 50 个蘑菇实例，你是用其中的 45 个来训练，用剩下的 5 个来做测试，还是 30 个用于训练，20 个用于测试？ 幸好**交叉验证**（cross-validation）可以帮我们跳过这个尴尬的境况，这是一种普遍适用的方法，它通过重复实验来预测模型的性能，而不是依靠数学分析。

交叉验证的基本思路是以不同的划分形式将原实例集多次划分成两部分，一部分用于训练，另一部分用于测试，再**重复**多次训练–测试实验，最后取测试结果的平均值。这一思路可以通过K **折交叉验证**来实现：将原集随机分为 K 个子样本，$K-1$ 个子样本用于训练，还有

一个子样本用于测试。重复这一过程 K 次，保证每个子样本有且仅有一次作为验证数据。最后，将各次的结果平均计算出一个估计值。这种方法的优势是每一个测点都既充当过训练数据，又充当过验证数据，并且刚好有一次用于验证。若实例集实在是非常小，则可使用交叉验证的一种极端情况——**留一验证**（leave-one-out cross-validation），每次留下原实例集中单独的一个测点用作验证数据，余下的实例都用作训练数据（这种情况下，K 等于实例数）。

分层交叉验证（stratified cross-validation）作为一种改进，可以避免训练集和测试集中不同类的平衡问题。它能够避免有时发生这种情况，即某一个类在训练集中很多，而在验证集中很少（相对于所有实例的平均出现率）。应用分层能够分别从每个类别中抽取出 ℓ/K 个测试样本，以保证不同类别的实例分布均衡（见图 3-3）。

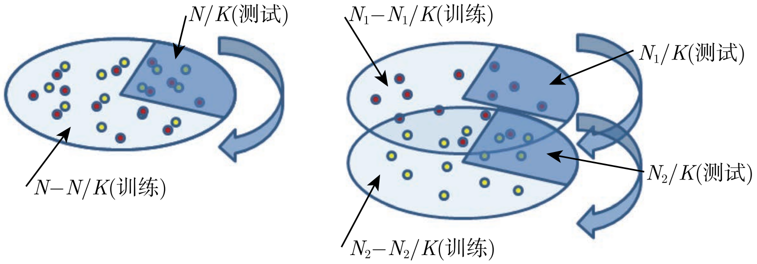


图 3-3　分层交叉验证，图中是两个类别的例子。在普通的交叉验证中，实例的 $1/K$ 用于测试，然后该切片“轮换”K 次。在分层的情形中，每个类别中有一个独立切片，用于保持两个类别的相对均衡

如果机器学习方法本身也有一些**需要调节的参数**，那么将产生一个附加问题。为了避免与模型中的基本参数混淆，或者说为了避免与多功能盒子中自定义的权重相混淆，我们称这些参数为**元参数**。假设我们想要确定一个迭代的最小化方法的终止条件（什么时候停止训练），或者一个多层感知器中的隐藏神经元的数量，再或者一个支持向量机（SVM）中的关键参数的合适取值。为元参数寻找最优值意味着需要多次**重用**验证集。而重用验证集又意味着它们也成为了训练过程的一部分。事实上我们正在处理一种元学习，也就是要学会学习的最佳方法。验证集被重用得越多，测得的性能就越可能过于乐观，这很危险，因为已经和新的数据上的真实表现不一致了。这就是现实版的“对数据进行的严刑逼供　　如果拷打得足够久，它会向你坦白任何事情”。

在上面提到的方法中，有限的实例集中的每一个实例都被用于各种用途，而合理的做法是，需要将数据分为 **3 个集合：一个训练集、一个验证集和一个（最后的）测试集**。其中测试集仅在最后测试性能时用到一次。

最后，请注意，在标准的单轮训练–验证循环中，“验证”和“测试”常被用作同义词，这可能会更令人感到困惑。

3.3 不同类型的误差

在测试一个模型的性能时，各种各样的误差带来的影响并不一样。如果你将有毒的蘑菇当作可食用的，你可能会有生命危险；如果你将可食用的蘑菇当作有毒的，你只是浪费了一点时间。根据问题的不同，确定最佳分类的标准也随之改变。考虑一个二元分类（输出“是”或者“否”）。一些可能的标准是：**准确率**（accuracy）、**精确率**（precision）和**召回率**（recall）。虽然它们的定义都很简单，但是需要小心区分以避免混淆（见图 3-4）。

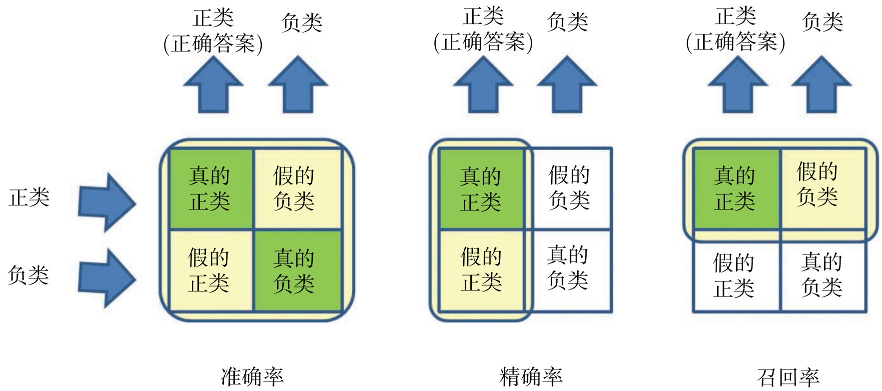
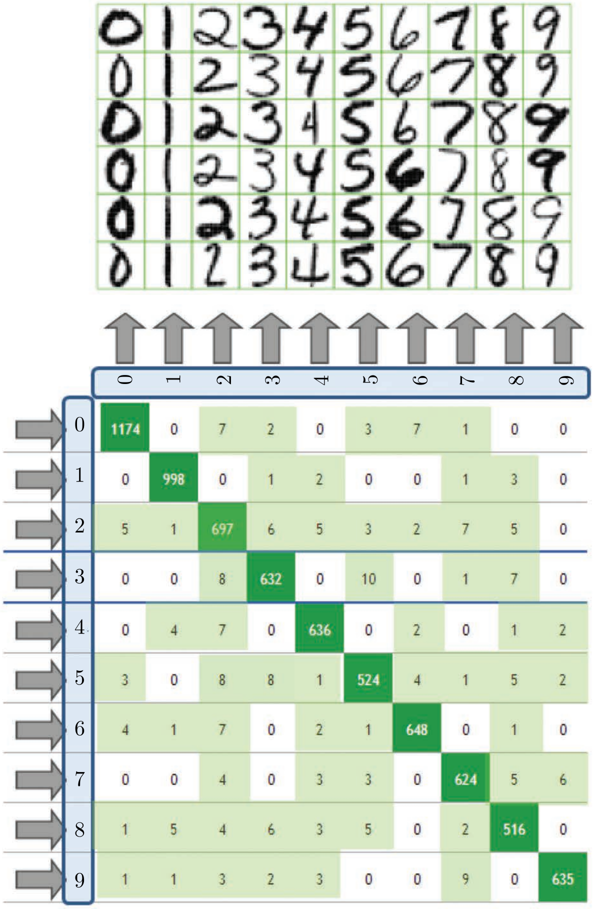


图 3-4 矩阵中的每一行报告一个类别的不同分类。你可以想象从左边进入，按不同的列进行分类，再从顶部退出的情况。准确率定义为正确答案占总体的比例，精确率定义为正确答案占标记为正类的比例，召回率定义为正确答案占本身是正类的比例。图中依据浅灰色区域的案例，对暗灰色区域的案例进行了划分

准确率是这个分类器给出正确结果（真的正类和真的负类）的比例。其他的度量标准都专注于被标记为属于此类（“正类”）的情况。**精确率**等于真的正类数（正确地被标记为属于正类的实例数）除以被标记为属于正类的实例数（真的正类数和假的正类数之和，假的正类是指错误地被标记为属于正类的实例）。**召回率**等于真的正类数除以本身属于正类的实例数（即真的正类数和假的负类数之和，假的负类是本应该标记为正类，却错误地被标记为负类的实例）。精确率回答这个问题：“有多少被标记为正类的案例是正确的？”召回率回答这个问题：“有多少正类的案例被正确地检索为正类了？”那么现在，在采摘蘑菇时，你是希望精确率更高，还是召回率更高？

混淆矩阵（confusion matrix）展示了不同案例的分类情况，其中有正确的分类，也有被混淆成其他类别的（见图 3-5）。

其中的每一行都展示了某个类别的情况：考虑的总实例数，以及其中有多少被正确地识别（对角线上的单元格），或者有多少被错认为其他类别的成员（其他列上的单元格）。



	0	1	2	3	4	5	6	7	8	9
0	1174	0	7	2	0	3	7	1	0	0
1	0	998	0	1	2	0	0	1	3	0
2	5	1	697	6	5	3	2	7	5	0
3	0	0	8	632	0	10	0	1	7	0
4	0	4	7	0	636	0	2	0	1	2
5	3	0	8	8	1	524	4	1	5	2
6	4	1	7	0	2	1	648	0	1	0
7	0	0	4	0	3	3	0	624	5	6
8	1	5	4	6	3	5	0	2	516	0
9	1	1	3	2	3	0	0	9	0	635

图 3-5　光学字符识别的混淆矩阵（手写的邮政编码数字）。其中的各种混淆也有道理可言。例如数字“3”被正确识别 632 次，被错认为“2”8 次，被错认为“5”10 次，被错认为“8”7 次。“3”从没有被错认为“1”或“4”，因为它们的形状差别太大了



梗　概

机器学习（ML）的目标是用一个训练实例集来建立系统，这个系统能够正确地泛化到新实例上，这些新的实例是在学习阶段没有见过的，但来自同一个问题。

ML 的学习即是为一个灵活的模型找到合适的参数值，这些参数要使得实例集上的**误差度量自动最小化**，同时也需要避免复杂的模型，从而增加正确泛化的概率。

这个系统的输出值可以是一个类（分类问题），或者是一个数值（回归问题）。在某些情况下，为了增加可用性，可以输出某一类的概率。

只要我们有丰富的有代表性的数据，我们可以在不知道背景知识的情况下建立一个准确的分类器。相较于基于专业领域知识的手动构建的系统，这是一个了不起的改变。

ML 是非常强大的，但是它要求严格的方法（一种 ML 的“教育学”）。可以肯定的是，**不要在训练集上测试性能**，因为这是弥天大罪：重用验证数据将导致过于乐观的估计。如果实例非常稀缺，你可以使用交叉验证这一手段来炫耀你是个 ML 专家。

为了安全起见，也为了置身于 ML 的天堂，你应该保留一些实例用于测试，仅在最后测试性能的时候使用它们。

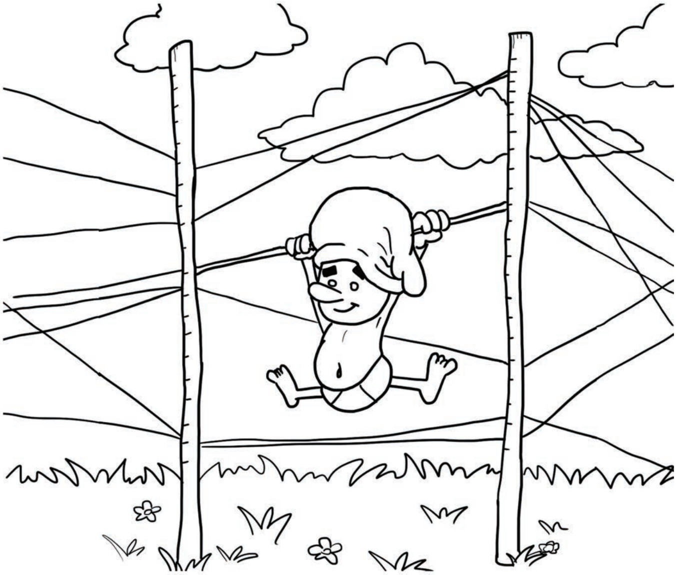
测试一个模型的性能的方法并不是唯一的，不同类型的误差可能造成不同的损失。**准确率**、**精确率**和**召回率**是二元分类中性能度量的一些可能的选择，对于更多类别的情况，一个**混淆矩阵**可以给出全部信息。

第一部分

监 督 学 习

第4章　线性模型

大多数惯用右手的人拥有线性思维，爱用传统的方式思考。(读者可以自由选择是否相信我们的开场白。)



优化的强大力量建立在拥有神奇力量的线性代数上。你是否记得在学校里老师说“好好学习线性代数，你会受益终身”? 好吧，多年以后你会发现他是对的。线性代数是“数学生存工具包”，当你面临一个棘手的问题时，应该首先试试线性方程组。在很多情况下，即使你不能用线性代数直接解决这些问题，至少也能得到一个不错的逼近。这不足为奇，解释数据的模型也是这样的。

图 4-1 中画出了不同车型的价格与它们的功率之间的函数关系。正如你所能想到的，功率越大的汽车，价格也会越高。汽车经销商是诚实的，这两个量之间有着近似的线性关系。如果用这个线性模型（这条直线）来总结这些数据，我们的确会损失一些细节，但是趋势会保留下来。我们所做的就是用直线来拟合（fitting）这些数据。

当然，定义我们所说的“最优拟合”会马上将我们引向优化（optimizing）对应的收益函数。

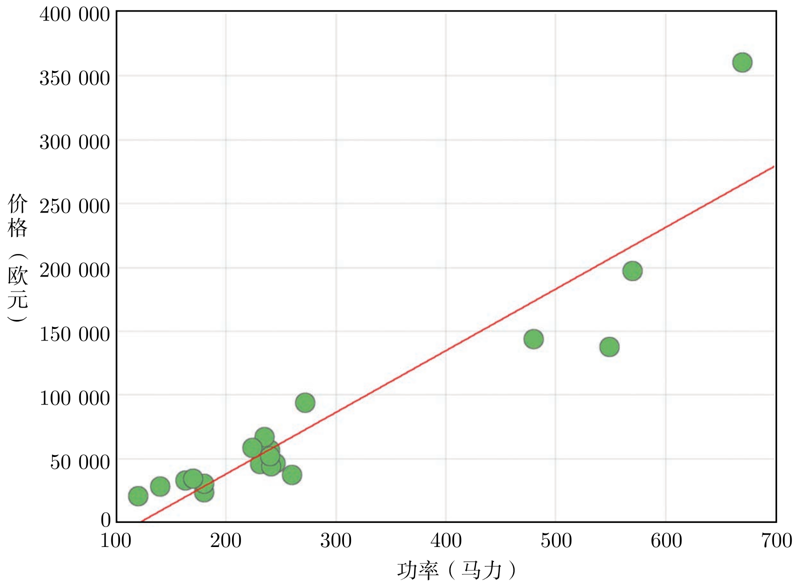


图 4-1 不同车型的价格与功率的数据。线性模型看起来像一条直线

4.1 线性回归

输入与输出特征的线性相关是一个广泛采用的模型。这一模型十分简单，并且训练起来很容易。另外，模型中每一项的权重系数都为这一项对应的特征的重要性提供了直观的解释：某一项权重系数的绝对值越大，对应的属性的影响就越大。所以，为了不让问题变得复杂，不要轻易尝试非线性模型，除非你的理由十分充足。

数学家不愿意浪费树木和纸张[①]，数组（向量）常用一个字母来表示，比如 $\boldsymbol{w}$。向量 $\boldsymbol{w}$ 由它的分量 $(w_1, w_2, \cdots, w_d)$ 组成，其中 d 是输入属性的个数，或者说维度。向量以一列的方式“站着”，为了使它们躺下来，你可以对它们进行转置操作，得到 $\boldsymbol{w}^{\mathrm{T}}$。因此，向量 $\boldsymbol{w}$ 和 $\boldsymbol{x}$ 之间的标量积就是 $\boldsymbol{w}^{\mathrm{T}} \cdot \boldsymbol{x}$，根据标准的矩阵乘法，相当于 $w_1x_1 + w_2x_2 + \cdots + w_dx_d$。

输出与输入参数成线性关系的这一假设可以表示为：

$$y_i = \boldsymbol{w}^{\mathrm{T}} \cdot \boldsymbol{x}_i + \epsilon_i,$$

其中 $\boldsymbol{w} = (w_1, \cdots, w_d)$ 为待确定的权重向量，ϵ_i 是误差项。在大多数情况下，假设误差项 ϵ_i 遵从高斯分布。即使一个线性模型能正确地解释这些现象，误差仍然会在测量时产生：每

① 在本书中我们不能非常深入地讲解线性代数，我们将给出基本的定义和动机，你能够非常容易地在专业书籍或者网站上找到进一步的内容。

一个物理量都只能在有限精度内测量。逼近并不是疏忽的产物，而是因为测量本身就不是完美的。

现在人们寻找一个权重向量 $\boldsymbol{w}$，使得线性函数

$$\hat{f}(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}} \cdot \boldsymbol{x} \tag{4.1}$$

尽可能逼近实验数据。这一目标可以通过寻找使平方误差和最小的 $\boldsymbol{w}^*$ 来达到（**最小二乘**逼近，least squares approximation）：

$$\text{ModelError}(\boldsymbol{w}) = \sum_{i=1}^{\ell} (\boldsymbol{w}^{\mathrm{T}} \cdot \boldsymbol{x}_i - y_i)^2 \tag{4.2}$$

如果在一个不太现实的场景里，测量误差为零并且有一个完美的线性模型，那么就留下了一个**线性方程**的集合 $\boldsymbol{w}^{\mathrm{T}} \cdot \boldsymbol{x}_i = y_i$，其中每一个方程对应一次测量，如果这个方程组是良好定义的（d 个未知数对应 d 个非冗余的方程），那么它们可以用标准的线性代数方法求解。然而在所有的现实场景里，测量误差都是存在的，并且测量数 $(\boldsymbol{x}_i, y_i)$ 可以远远大于输入维度。因此人们需要寻找一个近似解，确保权重向量 $\boldsymbol{w}$ 使得式 (4.2) 对应的值——通常大于零——尽可能小。

使用线型模型其实不需要了解式 (4.2) 是如何被最小化的，优化的真实信徒可以放心地相信它解决线性模型问题的神奇手法。但如果你对此充满好奇，或有自虐倾向，或要处理某些规模非常大而且比较困难的情况，可以考虑读一下 4.6 节和 4.7 节。

4.2　处理非线性函数关系的技巧

线性代数的美味现在肯定让你食欲大增了，但很可惜，并不是所有情况都能被一个线性模型解决。在很多情况下，一个形如 $f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$ 的函数是不实用的，因为它确实有着过多的限制。尤其考虑到它还假设 $f(0) = 0$。这个问题可以通过加入一个常数项 w_0 来解决，这样就从线性（linear）模型变成了仿射（affine）模型：$f(\boldsymbol{x}) = w_0 + \boldsymbol{w}^T \cdot \boldsymbol{x}$。这一常数项也可以并入到内积中，只需要重新定义 $\boldsymbol{x} = (1, x_1, \cdots, x_d)$，这样等式 (4.1) 对仿射模型仍然成立。

加入一个常数项是建模非线性函数关系方法的一种特殊情况，然而其他部分还属于最小二乘逼近的简单情况。这一明显的矛盾可以用一个技巧来解决：仍然用线性模型，只不过将原输入数据 $\boldsymbol{x}$ 进行非线性转换得到非线性属性，并在其上应用线性模型。我们可以定义这样的一个函数集：

$$\phi_1, \cdots, \phi_n : \mathbb{R}^d \longrightarrow \mathbb{R}^n$$

它从输入空间映射到某个更为复杂的空间，使得我们可以用向量 $\boldsymbol{\varphi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_n(\boldsymbol{x}))$ 进行线性回归，而不是原始数据 $\boldsymbol{x}$。

举例来说，如果 $d=2$ 并且输入向量 $\boldsymbol{x}=(x_1,x_2)$，输出的二次相关可以通过如下的基函数（basis function）得到：

$$\phi_1(\boldsymbol{x})=1,\qquad \phi_2(\boldsymbol{x})=x_1,\qquad \phi_3(\boldsymbol{x})=x_2,$$

$$\phi_4(\boldsymbol{x})=x_1x_2,\qquad \phi_5(\boldsymbol{x})=x_1^2,\qquad \phi_6(\boldsymbol{x})=x_2^2$$

注意定义 $\phi_1(\boldsymbol{x})$ 是为了使函数中允许常数项的存在。上面描述过的线性回归的方法则可以用在经过这些基函数变换后的六维向量上，而不是原来的二维参数向量。

更精确地说，我们寻找如下的一个函数，它是权重向量 $\boldsymbol{w}$ 和属性向量 $\boldsymbol{\varphi}(\boldsymbol{x})$ 的标量积：

$$\hat{f}(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\cdot\boldsymbol{\varphi}(\boldsymbol{x})$$

输出是这些变换后的属性的加权和。

4.3 用于分类的线性模型

4.1 节考虑了能近似拟合观察数据的线性函数，例如最小化平方误差的总和。然而在某些任务中，输出的可能取值被限定在一个很小的集合里。其中涉及分类问题。

假设输出变量是二值的（如 ±1）。在此情况下，线性函数可以用作**判别器**（discriminant），基本思路是让一个垂直于向量 $\boldsymbol{w}$ 的超平面将这两类隔离开。平面是直线的一般化；同样，当维度大于 3 时，超平面就是平面的一般化。

训练过程的目标是找到最佳的超平面，使得属于同一类的实例在这个超平面的一边，而属于另一类的实例在另一边。用数学语言表述就是，要找到最佳的系数向量 $\boldsymbol{w}$ 使得决策程序

$$y=\begin{cases}+1 & 如果\ \boldsymbol{w}^{\mathrm{T}}\cdot\boldsymbol{x}\geqslant 0\\ -1 & 其他情况\end{cases}\tag{4.3}$$

表现得最好。决定**最优线性分离函数**（best separating linear function，几何上的一个超平面）的方法取决于分类标准和误差度量的选择。

从这一章可知，如果要进行回归，可以要求第一类的点映射到 $+1$，第二类的点映射到 -1。这是一个比可分离更强的要求，但是让我们能够使用回归的方法，像梯度下降法（gradient descent）和广义逆矩阵法（pseudo-inverse）。此外，最小二乘法不仅可以实现两个类别样本的分类（如果这两类样本是可分的），还可以让分类是**健壮的，分割的超平面离样本都很远**，这也是接下来的章节将会遇到的一个主题。通过强制模型的输出为 $+1$ 或 -1，加上平方误差惩罚项，可以避免得到过多的分割超平面，从而提升模型的稳定性。一个普通的分割超平面可能使得正负样本的输出都接近于 0（例如 $+1$ 样本的输出值为 0.000001，-1 样本的输出值为 -0.000001），这样当我们使用该模型对有噪声的测试数据进行分类时，就很容易得到错误的类别（见第 11 章，图 11-1）。

如果训练实例不能被一个超平面所分离，要么忍受一些训练误差，要么尝试使用前面建议的技巧，从原始数据中计算出一些非线性的属性，使得变化后的输入能够被分离。图 4-2 给出了一个例子，两个在 0-1 坐标上的输入，输出是对应的异或函数（XOR 函数，一个输入与另一个输入的或，但不都等于 1）。

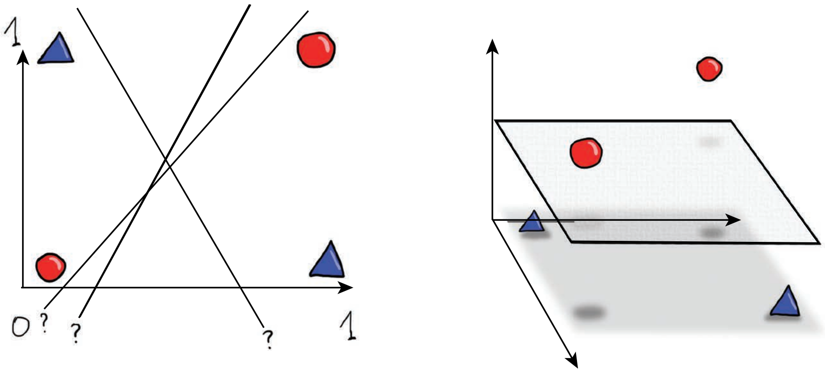


图 4-2　不能进行线性分离的情形（XOR 函数，左）。可以通过将点以非线性方式映射到更高维度的空间，通过超平面获得线性分离

在原二维输入空间里，这两类（以 1 或者 0 为输出）不能被一条直线（一个一维的超平面）分离。但是它们能够在转换后的三维输入空间里被一个平面分隔开。

4.4　大脑是如何工作的

我们的大脑是一团乱麻，至少本书作者的是这样。可以肯定的是，计算两个很大数的和的系统，与玩“赶尽杀绝”这类动作游戏的系统是很不一样的。进行逻辑演算或推理的系统认出母亲的脸的系统也是很不一样的。前一种系统是迭代的，它的工作方式是按照顺序的步骤来进行的，需要有意识地努力集中注意力。后一种系统以并行的方式工作，速度很快，无须太多努力，以非符号的方式（不会用到符号和逻辑）工作。

机器学习中的不同机制可以模仿这两类系统。线性判别器运用迭代梯度下降的学习方法来逐步改进，它更多模仿的是非符号系统；基于一连串“如果–那么–否则”规则（后面的章节将会提到它们）的分类树更多模仿的是逻辑系统。

用于分类的线性函数有许多名字，其中具有历史意义的一个是**感知器**，它强调了与生物神经元的类比。神经元以化学突触来通信（见图 4-3）。突触 [①]在神经元的工作中必不可少：神经元是一类特定的细胞，它们专门将信号传递给各个靶细胞，而完成这一工作需要依靠突触。

① “突触（synapse）”这个词是由希腊语“syn-”（在一起）与“haptein”（密切合作）合成的。

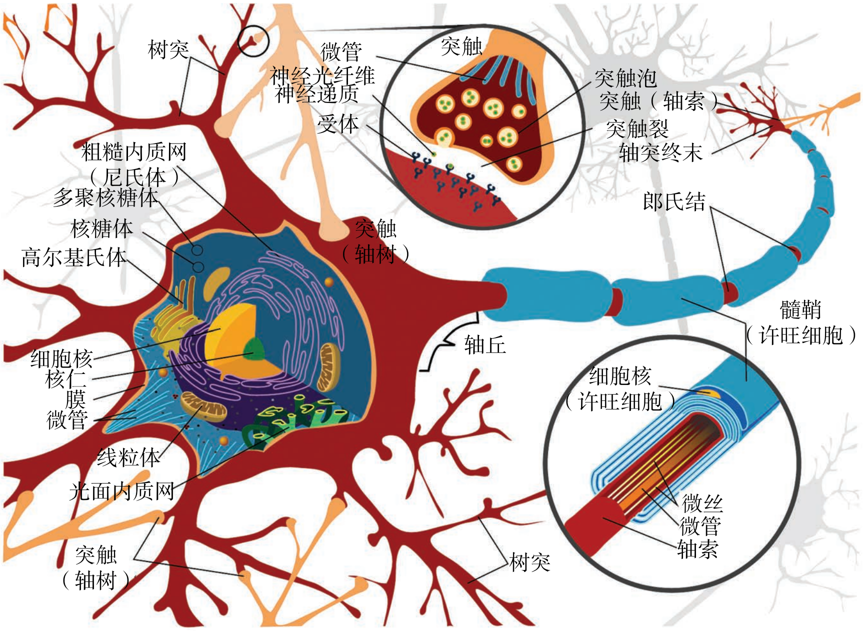


图 4-3　人类大脑中的神经元和突触

触发突触进行信号传递的基本过程是利用传播的电信号，这些电信号又是从神经元的电兴奋膜产生的。当且仅当兴奋性和抑制性突触的输入信号的结合超过某个给定的阈值时，这个电信号才会产生（神经元的输出打开了）。图 4-4 因此可以被看作单个神经细胞的抽象和功能性的表示。

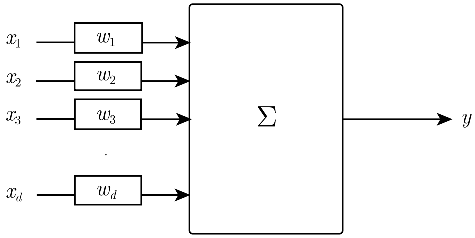


图 4-4　感知器：输入的加权和通过最终的“挤压”函数输出

4.5　线性模型为何普遍，为何成功

线性模型如此普遍的深层原因是存在于许多或大部分物理现象中的**平滑性**（“自然不允许跳跃”）。图 4-5 中的例子表明，青少年的平均身高随着年龄逐渐增长，而不是跳跃式地增长，直到青春期之后慢慢停滞。

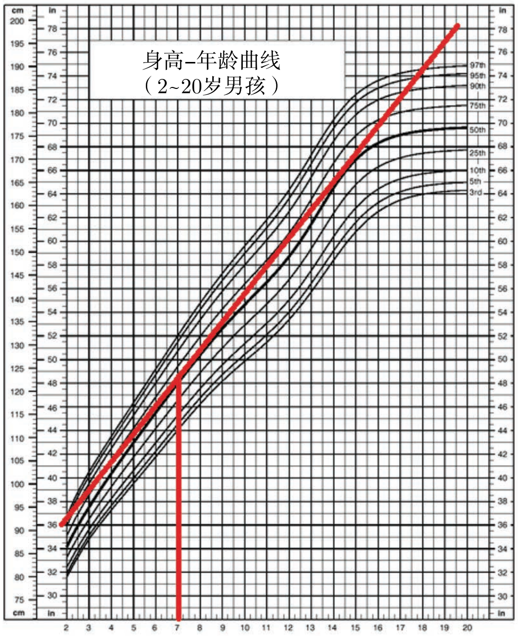


图 4-5　描述物理现象的函数一般都是平滑的。图中身高–年龄曲线可以用从 2~20 岁的切线逼近

现在，如果你记得微积分的知识，每一个光滑（可微）函数都可以在一个点 x_c 附近得到它的**泰勒展开式逼近**。这个展开式序列的第二项就是线性的，由梯度 $\nabla f(x_c)$ 和位移量的标量积给定，余项以二阶的速度收敛到零：

$$f(x) = f(x_c) + \nabla f(x_c) \cdot (x - x_c) + O(\|x - x_c\|^2) \tag{4.4}$$

因此，在平滑系统中，如果考虑与一个特定点 x_c 相距很近的点，那么线性逼近是一个合理的起点。

一般情况下，局部模型将只在一个给定的点附近表现得相当好。在青少年身高增长的线性模型中，7 岁那个点对应的切线在 15 岁以后就不适用了。幸好如此，否则我们的房子就不够大了。

4.6　最小化平方误差和

线性模型通过最小化式 (4.2) 中的平方误差和确定下来。如果你不满足于“证据在布丁里”①的说辞，而是想深入理解这件事情，那么请继续读下去。

① 这句谚语表明，“我”有足够的证据，但“我”不想论证了，你自己试试就知道了。（原意是，布丁好不好吃，得自己尝尝。）——译者注

前面提到过，在一个零测量误差和完美线性模型的不现实场景里，人们只需要解一系列**线性方程**$\boldsymbol{w}^{\mathrm{T}} \cdot \boldsymbol{x}_i = y_i$，每一个这样的方程对应于一次测量。如果这个系统是良好定义的（d 个非冗余的方程对应 d 个未知数），我们就可以通过对系数*矩阵求逆*（inverting the matrix）来解这些方程。

在实际操作中，让模型误差（ModelError）达到零是不可能的，另外数据点的个数会远远大于参数个数 d。并且，我们要记住学习的目标是泛化，我们感兴趣的是降低未来预测的误差。没必要过分地要求降低训练误差，以极低或零误差重新产生训练样本，这种要求事实上会适得其反。

我们需要**通过允许误差的存在来将线性方程组的解进行一般化**，从而一般化矩阵的逆。幸运的是，等式 (4.2) 是二次的，求它的最小值将再次得到线性方程组。实际上，你可能会意识到二次模型的成功正与这个事实有关：在我们计算导数后，就留下了一个线性表示。

如果你熟悉数学分析，那么求最小值很简单：计算梯度，然后要求它为零。如果你不熟悉数学分析，想想一个山谷的底部（最小值的那些点）的特征，就是一些小的移动，会使你保持在同一个高度。

下面的方程可以确定 $\boldsymbol{w}$ 的最优值：

$$\boldsymbol{w}^* = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\boldsymbol{y} \tag{4.5}$$

其中 $\boldsymbol{y} = (y_1, \ldots, y_l)$ 并且 X 是以向量 $\boldsymbol{x}_i$ 为行的矩阵。

矩阵 $(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}$ 是**广义逆矩阵**（pseudo-inverse），它是对那些非方阵矩阵的矩阵求逆的一种很自然的延伸。如果这个矩阵是可逆的，并且这个问题可以零误差地求解，那么广义逆矩阵就等于逆矩阵，但是一般情况下，例如训练实例数大于权重系数的个数时，寻求一个*最小二乘*解能避免无法找到精确解的尴尬，并且能提供一个统计上有意义的“折中”解。在现实中，精确模型与自然本身和物理测量的不精确性是无法兼容的，因此也难怪*最小二乘*和*广义逆矩阵*能跻身于最流行的工具中。

解等式 (4.5) 的方法是“一招制胜”：从实验数据中算出广义逆矩阵，然后相乘得到最优权重系数。在某些情况下，如果训练实例数非常大，**基于迭代方法的梯度下降**可能会更受欢迎：从一个初始权重系数开始，然后沿着负梯度的方向小步移动，直到梯度变为零，到达一个稳定点。顺便说一下，也许你已经想到了，真实的神经系统，比如大脑，并不采用线性代数这种“一招制胜”的方式，而是更多地使用迭代的方法，逐步地改进权重系数。也许这就是线性代数在学校里不那么受欢迎的原因。

注意平方误差的最小化有如图 4-6 中物理学上的弹簧模型的类比。想象一下，每一个样本点都有一根与刚性杆相连的垂直弹簧，这个刚性杆是最优拟合直线的物理实现。所有的弹簧都有一样的弹性系数，在松弛状态下长度为零。这种情况下，每根弹簧的势能与它长度的平方成正比，因此式 (4.2) 描述了这个系统的整体势能，仅相差一个常数因子。如果让这个物

理系统振荡起来，直到达到平衡，在阻尼振荡的情况下，那么这根刚性杆的最终位置就给出了最小二乘的拟合参数；这就是一台直线拟合的模拟计算机。

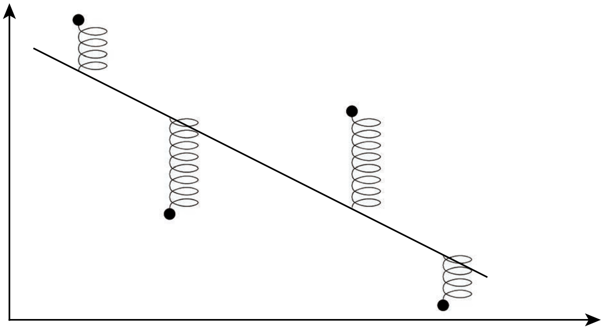


图 4-6　在物理学中，最小二乘拟合的原理类似于弹簧，最佳拟合的直线最小化整个系统中所有弹簧的势能（正比于弹簧长度的平方和）

你肯定会忘了广义逆矩阵，但肯定永远不会忘记这个阻尼振荡的弹簧物理系统，它能为这些实验数据找到最好的拟合直线。

如果属性是通过某个 φ 函数变形的（作为一个考虑非线性关系的技巧），求解的方式也十分相似。令 $\boldsymbol{x}'_i = \boldsymbol{\varphi}(\boldsymbol{x}_i), i = 1, \cdots, \ell$ 表示训练输入元组 $\boldsymbol{x}_i$ 的变形。如果 X' 是以 $\boldsymbol{x}_i$ 为行向量的矩阵，那么关于最小二乘逼近的最优权重系数可以这样求得：

$$\boldsymbol{w}^* = (X'^{\mathrm{T}} X')^{-1} X'^{\mathrm{T}} y \tag{4.6}$$

4.7　数值不稳定性和岭回归

实数（比如 π 和“大多数”的数）无法在数字计算机中表示出来，它们是“伪的”。数字计算机中的每个数都被赋以一个固定的有限的二进制数，没有方法来表示一个无限数位的数，像 3.14159265 . . .。因此，在计算机中表示的实数都是“伪的”，它们能并且经常造成误差。误差会在数学运算的过程中不断传播，在某些情况下，一连串运算的结果可能与数学上的结果相差甚远。找一个矩阵，求它的逆矩阵，并且将二者相乘。你期望会得到单位矩阵，但最后你却得到一个不同的答案。也许你应该查查银行使用的小数精度。

当训练实例数很大的时候，式 (4.6) 是超定（over-determined）情况的线性系统的解（线性方程多于变量）。特别是矩阵 $X^{\mathrm{T}}X$ 必须是非奇异的，这要求训练点集 $x_1, \cdots, x_\ell$ 没有全部落在 $\mathbb{R}^d$ 的某个真子空间里，也就是说它们没有被“对齐”。在很多情况下，即使 $X^{\mathrm{T}}X$ 是可逆的，也有可能因为训练点集的分布不那么合适而导致不稳定。**稳定性**在这里意味着样本点中的微扰只会造成结果中的微小改变。图 4-7 中给出了一个例子，如果样本点的选择不好（右图，x_1 和 x_2 不是独立的），将会使系统更多地依赖于噪声，甚至是舍入误差。

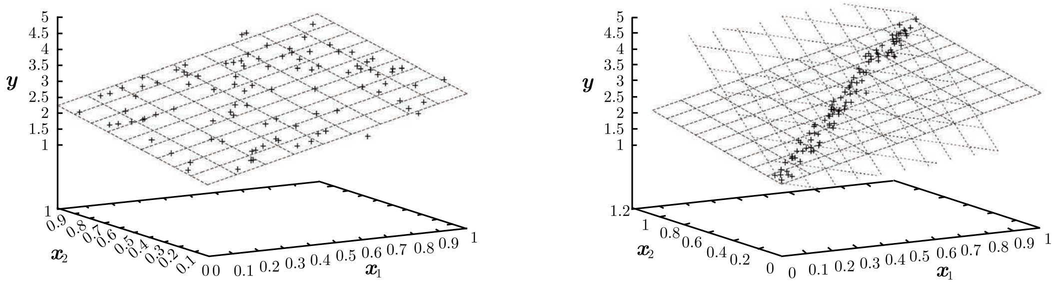


图 4-7 一个分布均匀的训练数据集（左侧）会得到一个稳定的数值模型，而一个糟糕的数据点选择将会得到许多差异很大的拟合平面，比如右侧图中的数据点几乎分布在一条直线上（引自参考文献 [9]）

如果没有办法来改变训练样本点的选择，而样本点又没有如愿地分布时，用以保证数值稳定性的数学工具是**岭回归**（ridge reghression）。它在需要最小化的（最小二乘）误差函数中加入了一个**正则化**（regularization）项：

$$\text{error}(\boldsymbol{w};\lambda)=\sum_{i=1}^{\ell}(\boldsymbol{w}^{\mathrm{T}}\cdot\boldsymbol{x}_i-y_i)^2+\lambda\boldsymbol{w}^{\mathrm{T}}\cdot\boldsymbol{w} \tag{4.7}$$

对 $\boldsymbol{w}$ 进行最小化，得到：

$$\boldsymbol{w}^*=(\lambda I+X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\boldsymbol{y}$$

在对角线上插入这些小的项使得求逆变得更加具有健壮性。另外，事实上人们也要求将权重向量的规模列入考虑范围，以避免出现如图 4-7 中右图那样陡峭的差值平面。术语“岭”（ridge）指的是将最优权重绘制为 λ 的函数时所造成的图形突起的模式。正如你所想象的，较大的 λ 值会导致总权重的收缩（图 4-8）。

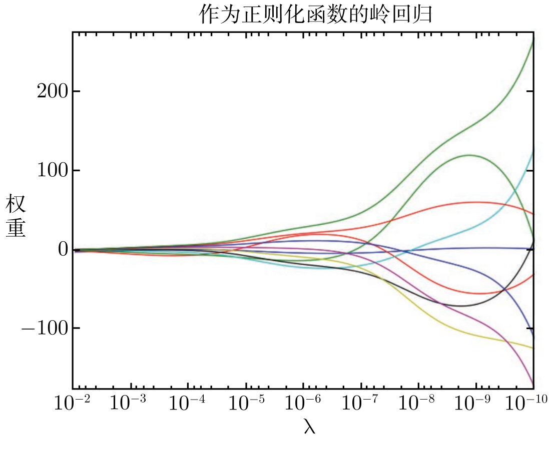


图 4-8 “岭”图，较大的 λ 值会导致总权重的收缩

如果你感兴趣的话，这一方法的理论基础是 Tichonov正则化，它是处理众多不适定（ill-posed）问题的最通用的方法。如果一个问题没有给出足够的信息来唯一确定一个解，那么这个问题被称为不适定的，例如实例数不够多。因此有必要提供一些额外的信息或者作出平滑性的假设。通过同时最小化实验误差和惩罚项，我们寻找这样一个模型，它不仅能很好地拟合，还足够简单，以避免在估计复杂模型时出现的大的变化。

你使用机器学习的时候不需要知道这些理论，但必须要意识到这个问题。当复杂的运算没有产生预期的结果时，这种意识将提升你的故障排除能力。避免非常大或非常小的数是一种解决大多数问题的实用方法，例如在机器学习之前，对输入数据进行缩放。



梗　　概

传统的线性回归模型（一组输入–输出对的线性逼近）通过**最小化线性模型预测值与训练样本输出值之间的平方误差和**来找到可能的最好的实验数据线性拟合。最小化可以是“一招制胜”，通过推广线性代数中的矩阵求逆，也可以通过迭代的方式逐步修改模型参数并降低误差。广义逆法可能是拟合实验数据的最常用的技术。

在分类中，线性模型旨在用线条、平面与超平面来分离实例。要确定分离平面，人们可以要求把输入值映射到两个不同的输出值（如 +1 和 −1）并使用回归。考虑到泛化性，找到健壮的分离超平面的更先进的技术是下面章节中将会描述的支持向量机。

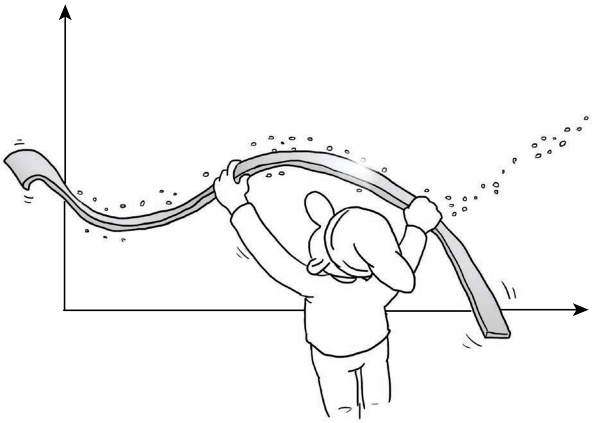
计算机中不存在实数，它们只能用有限大小的二进制数字逼近，而这可能会导致误差和不稳定（样本点的小扰动导致结果变化较大）。

一些机器学习方法与生物大脑从经验和功能中的学习方式存在松散的联系。学习骑自行车与符号逻辑和方程无关，而是关于如何进行逐步调整以及　　迅速从初始的事故中恢复过来。

第 5 章　广义线性最小二乘法

如无必要，勿增实体。

—— 奥卡姆的威廉（约 1285—1349）



上一章关于线性模型（模型的参数是线性的，即线性参数模型）的讨论留下了一些问题。一个严谨的建模工作的输出并不是一个单一的“带走它或者留下它”的模型。通常，人们通过**评价一个模型的性能**（拟合的优劣）来处理多种建模体系结构，通过**确定模型参数估计值的置信区间**（例如误差线）来**选择尽可能好的架构**，等等。读完本章之后，你应当可以从一个普通用户变成专业的最小二乘法大师。

上一章中提到了一个用于处理非线性性的技巧：用某个非线性函数 φ 对原输入进行映射，然后在转换后的输入空间里考虑一个线性模型（见 4.2 节）。虽然本章讨论的话题适用于一般情况，但是如果你记得单一输入变量的**多项式拟合**（polynomial fit）这一特殊情况，有助于你直觉上的理解。在单一输入变量的多项式中，非线性函数包含了原输入的幂，如下所示：

$$\phi_0(x) = x^0 = 1, \quad \phi_1(x) = x^1 = x, \quad \phi_2(x) = x^2, \cdots$$

这是让人特别感兴趣的一种方法，也在现实中被广泛应用，因此值得研究。

原始数据以成对值的方式给出：

$$(x_i, y_i), \qquad i \in 1, 2, \cdots, N$$

目标是推导出函数 $f(x)$，它要近似地建模 Y 对 X 的依赖关系，以便能在新出现的和未见过的 x 上计算函数值。

学习数据显著的模式和关系，意味着需要消除非显著的细节，例如测量*噪声*（由物理测量中有限精度导致的随机误差）。想想如何建模一个人的身高随着年龄改变的趋势。如果你重复用高精密仪器测量你的身高，那么每个测量会得到不同的值。这些带噪声的测量反映了一个简单的事实，那就是只能用有限的数位来描述你的身高（没有哪个精神正常的人会回答自己的身高是 1 823 477 微米）。

回到模型上来，我们并不要求函数也对噪声建模，也不要求函数图像准确地经过样本值（也就是说，不要求对所有的样本点都有 $y_i = f(x_i)$）。我们并不是做**插值**（interpolation），而是**拟合**（即兼容、相似或一致的）。不完全保真并不是一个坏处，相反还是一个优势，通过简化分析和允许忽略一些细节的推理论证，它能够提供建立更强大的模型的机会。如图 5-1 所示，一个插值数据集中所有点的函数与一个简单得多的函数进行比较，立刻可以显示出这些函数在建模数据分布上的行为是多么不同。*奥卡姆剃刀*说明了一个基本原则：相比于那些不必要的复杂的模型，简单的模型应该是首选。

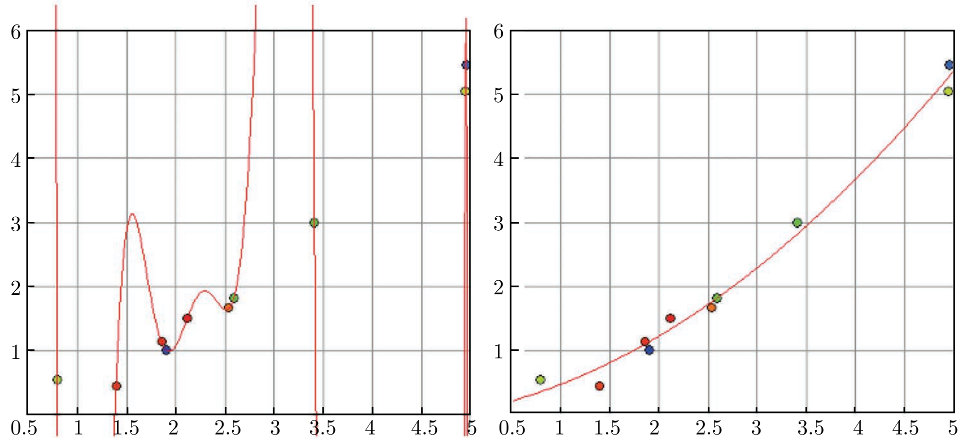


图 5-1　插值和拟合的比较。多项式的自由参数（等于其度数减 1）从数据点数目（左图）到 3 个（右图）的变化

享有选择不同模型的自由，例如选择不同次数的多项式，就应肩负起评价不同模型的优劣这一职责。评价多项式拟合的一个标准方法来自所得误差平方和的统计。

5.1　拟合的优劣和卡方分布

让我们以一个次数为 $M-1$ 的多项式开始，M 被定义为次数界（degree bound），等于次数加一。M 也是自由参数的个数（多项式中的常数项也算）。人们可以寻找一个合适次数的

多项式，它能够最好地描述这些数据分布：

$$f(x,\boldsymbol{c}) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{M-1} x^{M-1} = \sum_{k=0}^{M-1} c_k x^k \tag{5.1}$$

当与参数 $\boldsymbol{c}$ 的依赖关系不言自明的时候，为了简便，只写出 $f(x)$。因为一个多项式由它的 M 个参数（在向量 $\boldsymbol{c}$ 中）确定，所以只需寻找这些参数的最优值（opticmal value）。这是我们称为优化的力量的一个例子。一般的方法是：将问题形式化为函数最小化，然后借助优化的力量。

出于统计学和最大似然估计的考虑，如 5.2 节将要描述的那样，**卡方**（chi-square）评价函数被广泛应用于估计**拟合优度**（goodness-of-fit）。卡方是从希腊字母派生的名词，用以表示一个与其相关的统计分布 χ^2：

$$\chi^2 = \sum_{i=1}^{N} \left(\frac{y_i - f(x_i)}{\sigma_i} \right)^2 \tag{5.2}$$

如果参数 σ_i 都等于 1，那么解释很简单：这种情况下，χ^2 测量真实值 y_i 和模型估计值 $f(x_i)$ 之间的平方误差和，也就是前一章所描述的 ModelError($\boldsymbol{w}$)。

然而，在某些情况下，不同数据点的测量过程可能是不同的，人们对于某个测量的误差 σ_i 有一个估计，假设为标准差。试想用不同精度的仪器进行的测量，比如使用米尺和高精度卡尺。

对于米尺来说，毫米级的误差是可以接受的，但是对于卡尺来说，误差就要远远小于毫米级了，卡方的定义就是对这一事实精准的数学表达方式：当计算 χ^2 时，误差必须拿来跟标准差进行对比（即归一化，normalize），因此要将误差除以 σ_i。这样得到的结果是与实际误差规模无关的一个数，并且它的含义是经过了标准化的。

现在有了一个精确的用归一化的卡方来衡量多项式模型性能的方法，于是问题就变成了如何找到这些多项式系数来最小化这个误差。图 5-2 中给出了一个启发式的物理学解释。幸运的是，正如上一章所提到的，这个问题可以用标准的线性代数方法来解决。

这里用如下的方式来完成分析工作：取偏导数 $\partial\chi^2/\partial c_k$，并令其为零。由于卡方是系数 c_k 的二次函数，需要求解 M 个线性方程：

$$0 = \frac{\partial \chi^2}{\partial c_k} = 2 \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left(y_i - \sum_{j=0}^{M-1} c_j x_i^j \right) x_i^k, \qquad k = 0, 1, \cdots, M-1 \tag{5.3}$$

为了缩短这个数学表达式，可以方便地引入一个 $N \times M$ 矩阵 $A = (a_{ij})$，其中 $a_{ij} = x_i^j/\sigma_i$，包含了经过 σ_i 归一化之后的 x_i 的幂。再引入未知系数向量 $\boldsymbol{c}$ 和向量 $\boldsymbol{b}$，其中 $b_i = y_i/\sigma_i$。

很容易验证，式 (5.3) 中的线性系统可以写成以下紧凑形式：

$$(A^{\mathrm{T}} \cdot A) \cdot \boldsymbol{c} = A^{\mathrm{T}} \cdot \boldsymbol{b} \tag{5.4}$$

这称为最小二乘问题的法方程（normal equation）。

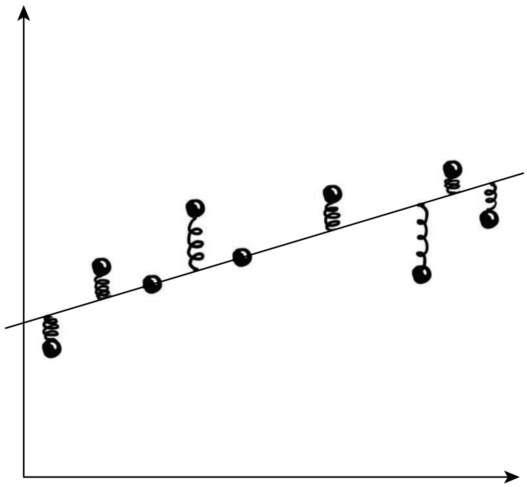


图 5-2 一条直线的拟合，有一个物理的类比：每个数据点与拟合直线间由弹簧连接。弹簧的强度正比于 $1/\sigma_i^2$。能量最低的情况对应于最小的 χ^2

令逆矩阵为 $C = (A^{\mathrm{T}} \cdot A)^{-1}$，则系数可以通过 $\boldsymbol{c} = C \cdot A^{\mathrm{T}} \cdot \boldsymbol{b}$ 得到。有趣的是，如果将系数向量 $\boldsymbol{c}$ 看作随机变量，那么 C 就是它的协方差矩阵：C 的对角线上的元素是拟合参数的方差（不确定度的平方）$c_{ii} = \sigma^2(c_i)$，而那些非对角线上的元素就是参数对之间的协方差。

矩阵 $(A^{\mathrm{T}} \cdot A)^{-1}A^{\mathrm{T}}$ 是**广义逆矩阵**，我们在上一章已经遇到过了，它将线性方程组的解从**最小二乘误差**的意义上进行了一般化：

$$\min \chi^2 \underset{\boldsymbol{c}\in\mathbb{R}^M}{=} \|A \cdot \boldsymbol{c} - \boldsymbol{b}\|^2 \tag{5.5}$$

若式 (5.5) 有一个精确解，则对应的卡方为零，并且拟合曲线刚好通过所有数据点。这种情况的发生要求我们有 M 个参数，还要有 M 个不同的数据点对 (x_i, y_i)，这是一个有 M 个线性方程和 M 个未知数的可逆系统。这种情况下，我们处理的就不是逼近拟合了，而是插值。如果没有一个精确解，就像数据点的个数多于参数的情况，广义逆矩阵给出了一个向量 $\boldsymbol{c}$，这一向量使得从欧氏范数的角度来说，$A \cdot \boldsymbol{c}$ 离 $\boldsymbol{b}$ 最近，这是对于近似解的一个非常直观的描述。记住，对于有噪声的数据，一个好的模型应该是能很好地归纳观察到的数据，而不是精确地重新产生它们，因此参数的个数必须（远）小于数据点的个数。

上面的推导不仅仅适用于多项式拟合，我们现在还可以非常轻松地拟合很多其他类型的函数。尤其是，如果这个函数是一些基函数 $\phi_k(\boldsymbol{x})$ 的线性组合，比如：

$$f(x) = \sum_{k=0}^{M-1} c_k \phi_k(\boldsymbol{x})$$

那么很多工作都已经完成了。事实上，将基函数的值 $a_{ij}=\phi_j(\boldsymbol{x}_i)/\sigma_i$ 替换矩阵 A 中对应的值就可以了。因此，我们现在有了一个拟合复杂函数的有效方法，例如：

$$f(x)=c_0+c_1\cos x+c_2\log x+c_3\tanh x^3$$

注意，这些未知参数必须以线性的方式出现，而不能出现在这些函数的参数中。例如，我们没法用这种方法拟合 $f(x)=\exp(-cx)$，或者 $f(x)=\tanh(x^3/c)$。我们最多可以尝试将这些问题转化成恢复基函数的线性组合的问题。第一种情况下，举例来说，我们可以用 $\hat{y}_i=\log y_i$ 拟合一个线性函数 $f(\hat{y})=-cx$，但是这个技巧在一般情况下却起不了作用。

图 5-3 中展示了散点图和对应的一条多项式拟合的曲线，这个多项式是二次的（一条抛物线）。这一曲线和数据点仅凭视觉上的比较就已经可以给出拟合优度（卡方值）的一个估计了。“目测卡方”（chi-by-eye）这一方法就是观察多项式拟合的图像，并且根据原数据的散点图来判断这一拟合好还是不好。

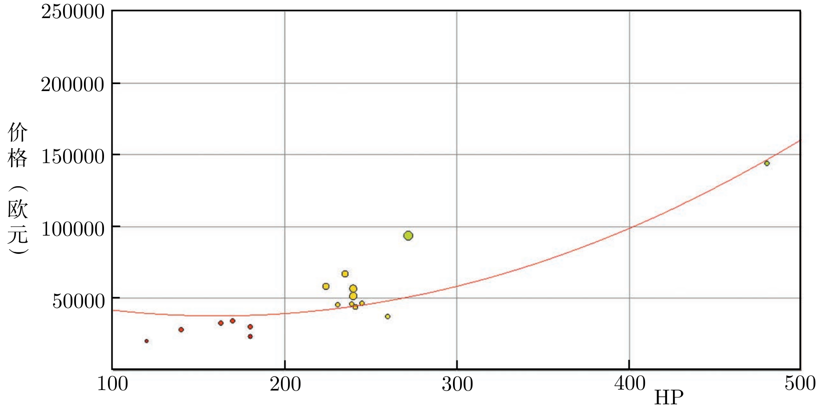


图 5-3 多项式拟合：作为发动机功率函数的车价

当实验数据不遵循多项式的规律时，用多项式来拟合并不是很有效，甚至很可能产生误导。正如前面所阐述的，通过增加多项式的次数来降低卡方值还是可行的：这将给模型更多的移动自由度，使得它尽可能接近实验数据点。事实上，如果参数个数等于数据点个数，这个多项式将会以零误差来插值这些点。但是这种减少误差的方式将造成原始数据点之间的曲线产生剧烈的振荡，如图 5-1（左）所示。这种模型并没有对数据进行归纳，它在泛化方面将面临非常大的困难。它没法对那些构建这一多项式之外的 x 值所对应的 y 值进行预测。

在统计学中，如果一个模型倾向于描述随机误差或噪声而不是数据间的基本关系，那么就会产生**过拟合**（over fitting）现象。当一个模型过于复杂时，例如相对于可用的数据量有太多的自由度（在我们多项式的例子中，就是有太多的参数），过拟合现象就会产生。

一般来说，过拟合模型的预测性能会很差。如果用人类的行为来打比方，可以想想教学：如果一个学生只关注并记住老师在课堂上讲的一些细节（例如数学课上某个特定练习的细

节），而不是提炼并理解基本的规则和意义，他只能靠着记忆空洞地重复老师的字眼，却无法将他的知识举一反三应用到新案例上。

5.2　最小二乘法与最大似然估计

了解广义最小二乘拟合的基本方法后，现在从统计学的角度来思考一些附加的动机。鉴于我们有选择不同模型的自由，比如拟合多项式的次数，那么用于辨别最佳模型构架的方法将是十分珍贵的，毕竟不能仅仅依靠肤浅的“目测卡方”方法。

下面是最小二乘拟合的过程。

(1) 假设大自然和实验程序（包括测量）会产生独立的实验样本 (x_i, y_i)。假设 y_i 的测量值受到了误差的影响，这个误差服从正态（即高斯）分布。

(2) 如果模型参数 $\boldsymbol{c}$ 是已知的，那么就可以估计我们测量数据的概率。在统计学的术语中，这叫作数据的**似然率**（likelihood）。

(3) 最小二乘拟合所找到的就是使得我们数据的似然率最大化的参数。最小二乘是一种**最大似然估计**（maximum likelihood estimator）。从直觉上来说，这使得选择的模型和观察到的数据之间的“契合度”最大化。

这种推演是很直观的。在此之前，你可能想复习一下高斯分布，5.3 节提供了相关内容。对于单个数据点，它的位置与测量值 y_i 的距离在区间 $\mathrm{d}y$ 中的概率正比于：

$$\exp\left(-\frac{1}{2}\left(\frac{y_i - f(x_i, \boldsymbol{c})}{\sigma_i}\right)^2\right)\mathrm{d}y \tag{5.6}$$

由于数据点是独立生成的，整个实验序列（似然性）的概率是单个概率的乘积：

$$\mathrm{d}P \propto \prod_{i=1}^{N} \exp\left(-\frac{1}{2}\left(\frac{y_i - f(x_i, \boldsymbol{c})}{\sigma_i}\right)^2\right)\mathrm{d}y \tag{5.7}$$

由于我们是求关于 $\boldsymbol{c}$ 的最大值，常数因子（像 $(\mathrm{d}y)^N$）是可以略去的。另外，最大化这个似然率等价于最大化它的对数（对数函数事实上是它的参数的一个递增函数）。好吧，由于对数的基本性质（即将乘积转换成求和、将幂转换成乘积，等等），当常数项被略去的时候，式 (5.7) 的对数就正好是式 (5.2) 中的卡方的定义。最小二乘拟合和最大似然估计之间的关系现在应该清楚了。

5.2.1　假设检验

统计学的假设检验可用来**评价**模型的**性能**。需要问的一个基本问题是：考虑这 N 个实验数据点，并且给定估计的 M 个参数值，大于等于我们测量的卡方的值有多大的可能性会碰巧出现？显然，这个问题将一个关于数据的问题（**“测量这些数据刚好得到这些测量值的**

似然率是多大”）用更加精确的统计学方式表达出来了，即**“从拟合这个模型的角度来看，另一个数据集比现有数据集更差的概率是多少”**。如果这个概率很高，那么从统计学的角度来说，y_i 和 $f(x_i, \boldsymbol{c})$ 之间的差异是有意义的。如果这个概率很低，要么你很不走运，要么你模型中的某些部分不起作用了：根据对大自然和测量生成过程的理解，测量值和期望之间的误差太大了。

令 $\hat{\chi}^2$ 表示在给定输入和输出集上所选的模型的卡方值。这个值服从一个被称为*自由度为 ν 的卡方*（χ^2_ν）概率分布，其中自由度 ν 确定数据集比模型“大”了多少。如果假设误差服从零均值和单位方差（记住，我们已经把它们归一化了）的正态分布，那么 $\nu = N - M$。一般情况下，正确的自由度还取决于表达误差分布（例如偏态）所需参数的数目。因此我们希望的拟合优度测量可以用如下的参数 Q 来表示：

$$Q = Q_{\hat{\chi}^2,\nu} = \Pr(\chi^2_\nu \geqslant \hat{\chi}^2)$$

对于一个给定的实验值 $\hat{\chi}^2$ 和给定的自由度，Q 的值可以计算，或在相关的表中查到[①]。

缩减的卡方统计量 χ^2_{red} 就是卡方除以自由度，在我们的例子中 $\nu = N - M$。缩减的卡方的优点是它已经规范化数据点的数量和模型复杂度。下面是一些经验法则。

- 如果 σ_i 是测量噪声的合理的估计并且模型也很合理，值 $\chi^2_{\text{red}} \approx 1$ 就是我们期望的。
- 如果 χ^2_{red} 的值太大，意味着你低估了你的误差来源，或模型拟合得不是很好。如果你信任你的 σ_i，也许增加多项式的次数可以改进结果。
- 最后，如果 χ^2_{red} 太小，那么模型 $f(x)$ 和数据 (x_i, y_i) 之间的契合会好得可疑。我们可能处于图 5-1 左图所示的情况。该模型“过度”拟合数据：模型不恰当地拟合噪声，或误差方差已被高估。我们应该试着减少多项式的次数[②]。

自由度 ν 会随着模型参数数量的增加而降低，在我们比较参数个数不同的模型时，它的重要性将变得明显。正如前文提到的，通过增加参数个数来降低*卡方*值是很容易的。使用 Q 值来测量拟合优度就把这个效应也考虑进去了。相对于一个误差很小但有大量参数的模型，*卡方*值更大（误差更大）的模型可能会产生更高的 Q 值（也就是更好）。

使用拟合优度 Q 度量，可以对不同的模型进行排名，然后选择最合适的那个。这一过程现在听上去是明确的，而且是量化的。如果你正在进行一个多项式拟合，现在可以用不同的次数来重复这一过程，测量 Q 并选择最合适的模型架构（最合适的多项式的阶）。

但魔鬼藏在细节里：*当假设正确的时候*，也就是 σ_i 是已知的、合适的并且独立的时候，这一机制才会是有用的。顺便提一下，对于不熟练的用户而言，要求 σ_i 值可能令人困惑。你

① 准确的公式为

$$Q_{\hat{\chi}^2,\nu} = \Pr(\chi^2_\nu \geqslant \hat{\chi}^2) = \left(2^{\frac{\nu}{2}}\Gamma\left(\frac{\nu}{2}\right)\right)^{-1}\int_{\hat{\chi}^2}^{+\infty} t^{\frac{\nu}{2}-1}\mathrm{e}^{-\frac{t}{2}}\,\mathrm{d}t$$

在 CPU 处理能力廉价的时代，这是非常容易计算的。

② 皮尔逊的卡方测试为评估拟合质量提供了客观的阈值，基于 χ^2 值、参数和数据点的数目以及所需的置信水平，如 5.2 节所述。

需要谨慎行事：**当假设错误时，统计就是一个雷区**，并且一个错误的假设可以使整个论证链被毁掉。

5.2.2　交叉验证

目前为止，我们展示了“历史上”的结果，统计学在计算机问世很久以前就诞生了，当时计算是十分昂贵的。幸运的是，如今丰富的计算资源使更强大的技术得以使用，这些技术能估计误差线，也能估计你的模型及其预测的置信度。这些方法不需要高深的数学，它们通常很容易理解，并且往往对于不同的误差分布都具有健壮性。

尤其是 3.2 节中提到的**交叉验证**方法，它可以被用来选择最佳的模型。和往常一样，基本的想法是把一些测量放在口袋里，用其他的测量来确定这个模型，然后将它们从口袋里拿出来估计新实例上的误差，重复几次并对结果求平均。如果数据足够丰富，这些泛化的估计可以是一个健壮的确定最优模型架构的方式。不同折数的交叉验证的结果分布提供了有关估计稳定性的信息，并允许断言，以给定的概率（置信度），预计的泛化结果将在一个给定的性能范围之内。推导性能估计的**误差线**的问题，或从更一般的层面上讲，为从数据中估计的任何数值推导误差线的问题，将在下一节探讨。

5.3　置信度的自助法

想象一下，大自然从一个真实的以 $\boldsymbol{c}$ 为参数的多项式中产生数据（输入–输出对）。大自然独立同分布地随机选择 x_i，并且根据式 (5.1) 和误差 ϵ_i 产生 $y_i = f(x_i, \boldsymbol{c}) + \epsilon_i$。

根据广义线性最小二乘法，你可以从提供的一对对 (x_i, y_i) 中确定最大似然值 $\boldsymbol{c}^{(0)}$。如果让大自然新产生一些数据，重复上面估值的过程，你没法保证会再一次得到同样的 $\boldsymbol{c}^{(0)}$ 值。相反, 你更有可能得到一个不同的 $\boldsymbol{c}^{(1)}$，然后是 $\boldsymbol{c}^{(2)}$，以此类推。

运行一次估值程序然后就用你所得到的 $\boldsymbol{c}^{(0)}$，这是不公平的。如果能将这一程序多运行几次，你可以得到系数的平均值，估计**误差线**（error bar），甚至可以使用不同的模型并且对它们的结果求平均（集成或者民主的方法将在下一章讨论）。误差线允许你量化估计中的置信度，这样你就可以说：以“90%（或者你决定的任何置信度）的概率，系数 c_i 是在区间 $c-B$ 到 $c+B$ 之间的一个值，B 是估计的误差线 ①”，或者以“99% 的置信度确信这个参数的真值在我们的置信区间内”。当使用模型的时候，类似的误差线可以在预测的 y 值上得到。对于仿真产生的数据，这种重复和随机化的过程被称为**蒙特卡罗实验**。蒙特卡罗方法是一类计算算法，依赖于重复随机抽样，以获取数值计算结果，即通过运行多次模拟，就像播放和录制在一个真实的赌场里的情况。这个名称来自摩纳哥公国的蒙特卡罗镇，它是拉斯维加斯的欧洲版本。

① 作为一个侧面观察，如果知道误差线是 0.1，你会避免小数点后的数字太多。如果估计你的身高，请不要写“182.326548435054 厘米”，精确到 182.3 厘米（加上或减去 0.1 厘米）足矣。

另一方面，大自然，即生成数据的过程，只能提供一组测量数据，由于反复测量代价过于昂贵而无法负担。只对同一组数据采用不同的随机估计，怎样才能尽可能利用好呢？一开始，它看起来是荒谬而不可能的任务。类似的荒谬曾出现于“令人惊讶的奇遇记”里，敏豪生男爵提着头发把自己和马从水坑里拉起来（见图 5-4），为了效仿他，可以尝试“拽着提鞋带（bootstrap）把自己拉过篱笆”，因此术语*自助法*（bootstrapping）的现代意义是描述一个自给自足的过程。



图 5-4 敏豪生男爵提着头发把自己和马从水坑里拉起来

好吧，结果还真有一种使用一组测量就能模仿真实的蒙特卡罗方法的手段。它可以通过构建一些观测到的数据集的（同样大小的）*再抽样*（resample）来实现。每一个新的样本可以通过*随机有放回抽样*（random sampling with replacement）来得到，这样同一个实例可能被多次取到（见图 5-5）。根据简单的数学知识，对于很大的实例数 N，大约有 37%（实际上大约为 1/e）的实例不会出现在一个抽样里，因为它们都被某些原始实例的多个副本代替了①。

对于每个新的第 i 个再抽样，重复拟合过程，会得到很多模型参数的估计值 $\boldsymbol{c}_i$。然后人们可以分析各个估计是如何分布的，使用观察到的频率来估计一个概率分布，并且总结出一个带置信区间的分布。例如，将置信水平固定在 90% 之后，人们可以确定 $\boldsymbol{c}$ 的中位数周围的一个区间，一个估计的 $\boldsymbol{c}$ 以 0.9 的概率落在这个区间内。视复杂程度而定，一维以上的置信区间可以是一个长方形的区域或一些更具灵活性的区间，如椭圆形。图 5-6 给出了一个一维（需要估计一个单一参数 c）的置信区间的例子。注意，对于任意的分布都可以求得置信区间，

① 尽管有着蛮力的快速上手的外表，自助法在统计学家中越来越享有声誉。其基本思想是，实际的数据集被视为包含一组狄拉克脉冲（Dirac delta）函数的某概率分布上的测量值，在多数情况下是原概率分布的最优估计 [88]。

不必非得是正态的不可，并且置信区间不一定是关于中位数对称的。

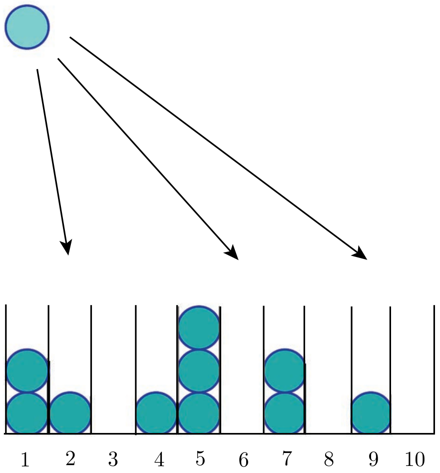


图 5-5　自助法：将 10 个球以均匀概率放在 10 个盒子里。自助法决定哪些实例和多少副本出现在自助样本里（实例 1 有两个副本，实例 2 有一个副本，实例 3 没有副本，等等）

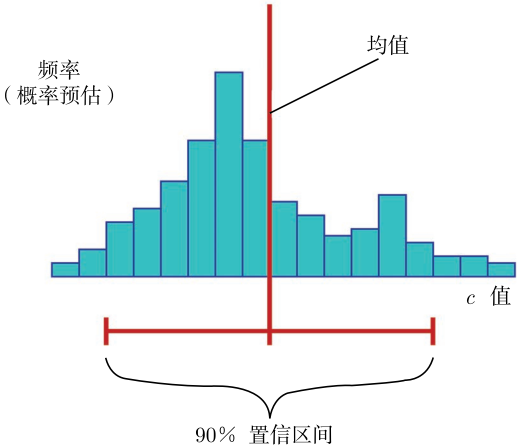


图 5-6　置信区间：从表征 c 估计值的分布的直方图中，可以推导出平均值区域周围 90% 的实例。也可以使用其他的置信水平，像 68.3%、95.4% 等（在正态分布的情况下对应于 σ 与 2σ 的历史概率值）

附录：绘制置信区间（百分位值和箱形图）

一个用于分析估计参数分布的快速上手方法是使用直方图（对在一个区间集合内的值的

频率进行计数)。在某些情况下,直方图包含多于所需的信息,并且这些信息不容易解释。一个用来刻画某个值的分布的紧凑方法是用它的**平均值** μ。给定一个包含 N 个值 x_i 的集合 $\mathcal{X}$,平均值是:

$$\mu(\mathcal{X}) = \left(\sum_{i=1}^{N} x_i\right)/N, \qquad x_{i,\cdots,N} \in \mathcal{X} \tag{5.8}$$

这个平均值被称为期望值,或者数学期望,或者均值,或者一阶矩(first moment),并且也用不同的记号来表示,例如 $\bar{x}$ 或者 $E(x)$。

一个相关但是又不同的数值是**中位数**,它的定义是将样本分成大的一部分和小的一部分的那个数。给定一个有限的数值列表,可以将所有这些观测值从大到小排列,中位数就是中间那个。如果有一些**离群值**(outlier,离大多数值非常远的数据),那么中位数比平均值更具有健壮性,平均值受这些离群值的影响很大。相反,如果数据点聚集在一块,就像由正态分布所产生的那样,平均值就趋向于与中位数重合。通过考虑**百分位数**,可以将中位数的定义一般化:一定百分比的观测值将小于这个变量值。那么第 10 百分位数就是有 10% 的观测值小于这个数。**四分位数**是一种特殊情况,包括下四分位数(第 25 百分位数)、中位数和上四分位数(第 75 百分位数)。四分位差(IQR),又称为中间离差(midspread)或者中间五十(middle fifty),是一个统计分布的测量,等于第一个四分位数和第三个四分位数的差。

箱线图,又称为**盒须图**,通过展示 5 个数字来总结数值集:最小的观测值(样本最小值)、下四分位数(Q1)、中位数(Q2)、上四分位数(Q3)和最大的观测值(样本最大值)。箱线图同样可以指出某些观测值是离群值(如果有的话),通常用圆圈表示出来。在箱线图中,箱子的底部和顶部总是上下四分位数,靠近箱子中间的那一杠总是中位数。须的两端可以表示若干不同的值,例如:

- 数据的最大值和最小值;
- 数据中大于及小于平均值一个标准差的值;
- 第 9 百分位数和第 91 百分位数;
-

图 5-7 展示了一个箱线图和 1.5 IQR 的须,这是通常(默认)的值,当数据符合正态分布时,它对应着 $\pm 2.7\sigma$ 和 99.3% 的面积。换句话说,对于一个高斯分布,少于 1% 的数据将落在盒须的范围之外,这是一个识别可能的离群值的实用标志。前文提到过,离群值是一个显著偏离包含它的样本中其他成员的观测值。在任何分布中,离群值都有可能存在,但是它们通常要么表示测量误差,要么总体服从*重尾分布*(heavy-tailed distribution)。前一种情况下,应该丢弃这些离群值或者使用对离群值有*健壮性*的统计;在后一种情况下,人们需要谨慎对待依赖于正态分布假设的工具和直觉。

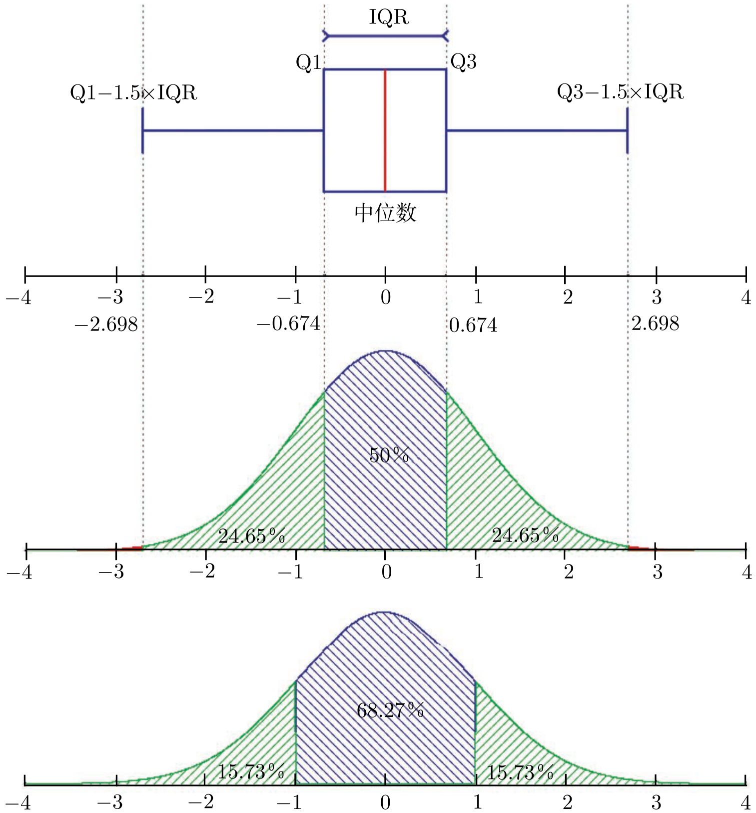


图 5-7 箱线图和正态分布之间的比较，横轴表示相对于标准偏差 σ 的位置。例如，在底部的图中，68.27% 的点落入距离平均值的一个标准差 σ 范围内



梗　概

多项式拟合以一种特定的方式使用**线性系数模型**（linear-in-the-coefficients model）来处理非线性问题。该模型包括（待定）系数的线性加权和乘以原始的输入变量的积。如果积被替换为输入变量的任意函数，相同的技术也可以使用，只要这个函数是固定的（函数中没有自由参数，仅作为乘法系数）。通过最小化平方误差来确定最优系数，这就意味着求解一组线性方程组。如果系数的数目大于输入–输出实例数，会出现**过拟合**（over-fitting），用这样的模型来推断新输入值的输出结果是危险的。

多项式拟合的优度（goodness of a polynominal fit）可以通过预测观察到与实测数据的差异的概率来评价（给定了模型参数后数据的似然率）。如果这个概率很低，那么不应该太过于信任该模型。但关于误差如何生成的错误假设容易导致我们得出过于乐观或过于悲观的结论。统计从假设开始建立坚实的科学建筑。如果建立在无效假设的沙土上，即使最坚实的统计建筑也会倒塌粉碎。幸运的是，基于可行性强的大规模计算的方法（例如交叉验证）是容易理解的，并且具有健壮性。

像**自助法**（bootstrapping）这样“荒谬”的方法（对同一数据进行带放回的再抽样，并以蒙特卡罗的方式重复估计过程），可以用于获取估计的参数值周围的置信区间。

你不过是最大化了自己被当成线性最小二乘法大师的概率。

第 6 章　规则、决策树和森林

如果森林中的一棵树倒下了，但是没有人听见，那么它是否发出了声音？



规则是一种**将知识模块用让人易于理解的方式结合起来**的方法。如果“客户很富有”，那么“他将会购买我的产品”。如果“体温超过 37 摄氏度”，那么“这个人生病了”。决策规则普遍应用于医疗、银行和保险领域，以及与客户打交道的特定流程中。

在一条规则中，人们区分**前件或前提**（一系列测试）和**后件或结论**。结论是对应输入并使得前提成立的输出类别；如果类别不能 100% 确定，就给出这些类别的概率分布。通常，这些前提都是用“且”连接起来的，也就是说，如果我们要“发射”这条规则，即得到结论，那么所有测试都必须通过。如果“距离小于 2 英里”且“晴天”，那么“步行”。一条测试可以针对类型变量的值（“晴天”），或者数值变量简单运算的结果（“距离小于 2 英里”）。如果想要让人理解，计算就必须简单。一个实用的改进是将前提为假时的分类也加入到同一个语句中。如果“距离小于 3 公里”且“没有私家车”，那么“步行”，否则“坐公交车”。

提取知识模块形成一个简单规则的集合是诱人的。但是**手动设计和维护规则是昂贵**且困难的。当规则集变大时，复杂性就会显现，就像图 6-1 中的规则会导致不同的甚至矛盾的分类。在这些情况下，分类可能与不同规则在数据上进行测试的排列顺序有关。从数据中**自动提取不矛盾的规则的方法**是很宝贵的。

相比处理带很多测试的非常长的前提，将规则分解成一串简单的问题更有价值。在贪心法中，那些信息量最大的问题最好放在这一序列的前面，这样可以**给问题加上层次结构**，从信息量最大的问题到信息量最小的问题。如上这些动机很自然的让我们考虑到**决策树**（decision

tree)，它是决策规则的有组织的分层排列，并且没有矛盾（见图6-2上图）。

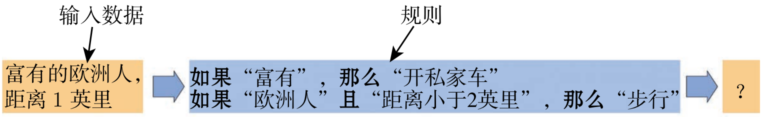


图 6-1　一套非结构化规则会导致矛盾的分类

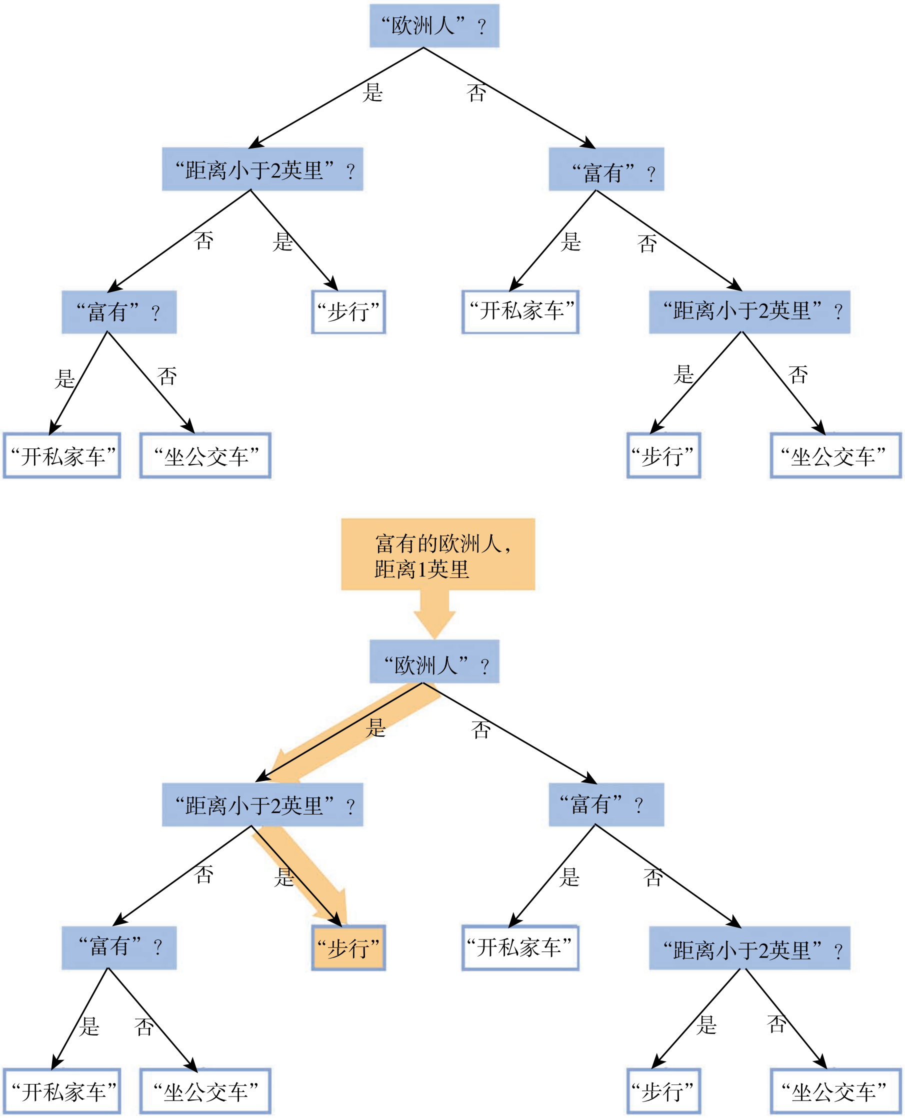


图 6-2　决策树（上图），以及同一棵树中达到分类的一种情况（下图）。抵达拆分节点的数据点根据测试函数的结果被发送到其左子节点或右子节点

决策树在机器学习（ML）诞生之初就非常普及了。现在，确实只有小而浅的树能够被人们“理解”，但是近年来由于计算和存储能力的极大提升，决策树的普及度越来越广了。很多树（在某些情况下可达到上百棵）可以组合使用，称为**决策森林**（decision forest），来得到具有健壮性的分类器。当考虑森林时，关心人们是否能理解这一问题被移至幕后，务实地寻找没有过度训练风险的、具有健壮性的高性能表现成为台前的主角。

6.1　构造决策树

决策树是以层次的方式组织起来的一个问题集，并且用一棵树的图形来表示。由于历史的原因，机器学习中的树，如同所有在计算机科学领域中的树那样，常常把根画在上方——如果你在北半球，那就想象澳大利亚的那些树。对于一件给定的事物，决策树通过连续地提出关于其已知属性的问题来估计它的一个未知属性。下一个问题问什么取决于前一个问题的答案，如果用图形来表示，这一事物对应于这棵树中的一条路径，如图 6-2 下半部分的粗线条所显示的。决策则根据这条路径的终端节点来做出。终端节点称作叶子。决策树可以被看作将复杂问题分解为简单问题集的一种方法，分解过程在这个问题已足够简单，即已到达叶子节点并找到已有的答案时结束。

一个从已标记的实例来构造决策树的基本方法是按照贪心法来进行的：在层次结构中，信息量越大的问题就越靠前。考虑一下初始的被标记的实例集。一个有两个可能输出（“是”或“否”）的问题将这一实例集分为两个子集，其中一个子集包括答案为“是”的实例，另一个子集包括答案为“否”的实例。初始的实例集是混乱的，包含不同类的实例。当问完一串问题，从根到了叶子时，叶子中余下的集合就应该几乎是“纯”的了，也就是只包含同一类的实例。这一类别也就是所有到达这片叶子的实例所对应的输出。

我们需要从初始的混乱的集合过渡到最终一系列（几乎）纯的集合。一个瞄准这一目标的**贪心**法就是从“信息量最大的”问题开始。这会把初始的集合划分为两个子集，分别对应答案“是”或“否”，也是初始根节点的子节点（见图6-3）。贪心法将以尽可能接近最终目标这一原则迈出第一步。在贪心法中，第一个问题的设计要使得这两个子集尽可能纯净。第一次划分完成之后，以**递归**的方式（见图6-4），继续对左右两个子集使用同样的方法，设计合适的问题，如此重复，直到剩下的集合足够纯净，递归停止。完全的决策树是由一个从上到下的过程导出的，这一过程通过在创造的子集中实例的相对比例来引导。

决策树的两个最主要的组成部分，一是纯度的定量度量，二是每个节点问的问题的类型。我们都同意纯度最大值是在子集中只有一个类别的实例时取到，而不同的度量负责测量那些不只一个类别的情况。其他的组成部分与终止条件有关：记住我们的目标是泛化，因此我们不希望构造一棵非常大的树，而每片叶子只有一两个实例。某些情况下，我们允许在子集尚未达到十分纯净时停止训练，并且当一些实例到达某个给定的叶子节点时，输出一个概率值。

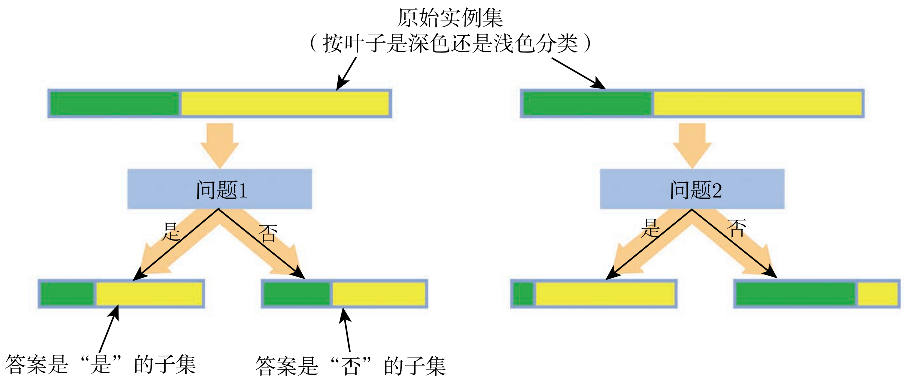


图 6-3 纯化集（两个类的实例）：问题 2 产生的子节点的实例子集更纯净

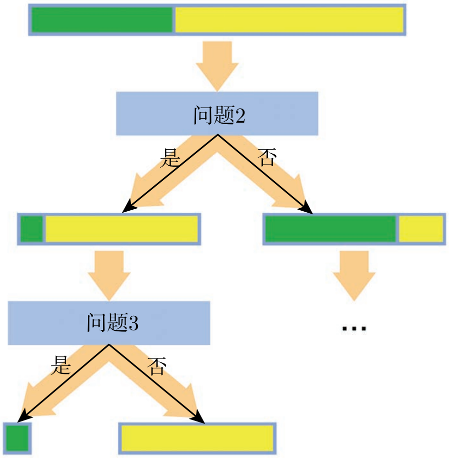


图 6-4 树构建中的递归步骤：经过问题 2 初步纯化后，将同样的方法应用在左边和右边的实例子集上。在这个例子中，问题 3 就足以完全净化这些子集。不需要在纯子集上执行额外的递归调用

在下面的描述中，假设所有涉及的变量都是类别变量（即名称，像上面例子中的“欧洲人”）。还有两种广泛使用的子集纯度度量，一是信息增益（information gain），二是基尼混度（Gini impurity）。注意我们处理的是有监督分类，因此我们知道这些训练实例的正确输出分类。

信息增益 设想我们从一个内部节点或叶子节点对应的集合中进行抽样。我们得到 y 类实例的概率 $\Pr(y)$ 正比于集合中该类实例所占的比例。所得类的统计不确定性由标记概率分布的香农**熵**来度量：

$$H(Y) = -\sum_{y \in Y} \Pr(y) \log \Pr(y) \tag{6.1}$$

在信息论中，熵量化了确定某件事件发生所需的平均信息（见图 6-5）。如果对数的底为 2，信息（也就是熵）的单位是二进制位（bit）。当所有 n 个类别的实例均分了一个集合时，熵就达到最大值，$H(Y) = \log n$；而当所有实例都属于同一类别时（这种情况下，不需要任何信息，我们已经知道我们会得到什么类别了），熵就达到最小值，$H(Y) = 0$。

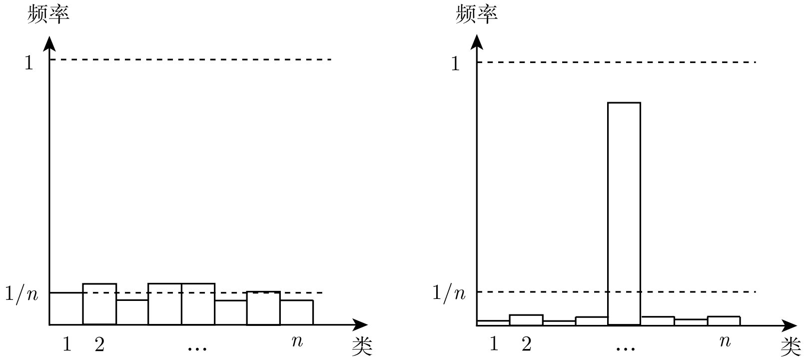


图 6-5　有着差别很大的熵的两个分布。高熵（左）：事件有相似的可能性，不确定性接近最高（$\log n$）。低熵（右）：事件有非常不同的可能性，不确定性非常小，接近零，因为某个事件占了大多数的概率

在信息增益方法中，**一个集合的混度由类别概率分布的熵来度量**。

知道一个问题的答案将会降低熵，或者熵保持不变，仅当答案不取决于类别时。令 $\mathcal{S}$ 表示目前的实例集，并且让 $\mathcal{S} = \mathcal{S}_{\mathrm{YES}} \cup \mathcal{S}_{\mathrm{NO}}$ 表示问过一个关于某属性问题之后的划分。理想的问题不会留下难以决策的情况：$\mathcal{S}_{\mathrm{YES}}$ 中所有实例是一类，而 $\mathcal{S}_{\mathrm{NO}}$ 中所有实例是另一类，因此这两个生成的子集熵为零。

在知道了答案后，熵的平均减少量被称为“信息增益”，它也是答案与类别变量之间的互信息（MI）。信息增益（互信息）可表述如下：

$$\mathrm{IG} = H(\mathcal{S}) - \frac{|\mathcal{S}_{\mathrm{YES}}|}{|\mathcal{S}|} H(\mathcal{S}_{\mathrm{YES}}) - \frac{|\mathcal{S}_{\mathrm{NO}}|}{|\mathcal{S}|} H(\mathcal{S}_{\mathrm{NO}}) \tag{6.2}$$

计算熵所需的概率可以用样本子集内各类别的相应频率来逼近。

信息增益是由 Quinlan 率先在 ID3、C4.5 和 C5.0 方法中使用的 [89]。值得注意的是，由于我们感兴趣的是泛化，信息增益对此是有用的，但并不完美。假设我们为描述某项业务客户的数据构造一棵决策树，每个节点可以有多于两个子节点。一个输入属性可能是客户的信用卡号码。因为这一属性唯一地标识每个客户，所以它与任何分类都有很高的互信息，然而我们不希望将其包含在决策树中：根据客户的信用卡号码来对客户作出相应决策，这一做法不太可能推广到之前没有见过的客户（这就是过拟合）。

基尼混度　试想一下，我们从一个集合中随机抽取一个元素，并随机贴上标签，概率正比于不同类别在这个集合中所占的比例。尽管这种方法看起来很原始，如果集合是纯的，这

种简单直接的方法的误差为零；如果某个类别在集合中占据了主要部分，这种方法的错误率也是很小的。

一般情况下，基尼混度（GI）测量从集合中随机选择的元素会被错误地标记的频率，标记是按照集合中标记的分布随机进行的[①]。它可以用错误率的期望来计算：对于每个类别，将该类别中某个元素被误分为其他类的概率（即将其分配给任何不正确类别的概率：$1-p_i$）相加，再乘以该元素属于该类别的概率 (p_i)，然后将这些乘积加起来。假设有 m 个类，并且令 f_i 表示集合中标记为 i 的元素比例。然后，通过频率来估计概率（$p_i \approx f_i$）：

$$\mathrm{GI}(f)=\sum_{i=1}^{m} f_i(1-f_i)=\sum_{i=1}^{m}(f_i-{f_i}^2)=\sum_{i=1}^{m} f_i-\sum_{i=1}^{m} {f_i}^2=1-\sum_{i=1}^{m} {f_i}^2 \tag{6.3}$$

当节点中所有实例都属于单一目标类别时，GI 为最小值（零）。GI 被用在 Breiman 提出的 CART 算法（分类回归树）中 [29]。

当考虑每个节点所问的问题类型时，实际上只需要考虑那些**有二元输出的问题**就足够了。对于类别变量，这个测试可以基于该变量的可能值的某个子集（例如，若某天是“星期六或星期天”，则回答 YES，否则 NO）。对于实值变量，每个节点对应的测试可以基于单一变量（例如，距离小于 2 英里）或者变量的简单组合，例如变量子集的一个线性函数与一个阈值进行比较（例如，客户花在汽车和摩托车上的钱的平均值多于 2 万美元）。上述概念可以被推广到要预测的数值变量，发展为**回归树**[29]。每片叶子要么包含到达这里的所有实例的平均输出值，要么包含从这些实例推导出的一个简单模型，比如线性拟合（后一种情况称为**模型树**）。

实际的数据中，**缺失值**就像冬天的雪花漫天飞舞。缺失值有两种可能的意义。在某些情况下，一个值如果缺失了，它会提供一些有用的信息（例如，在市场营销中，如果一个客户没有回答某个问题，我们可以假设他对此不是很感兴趣），缺失值可以被当作一个类型变量的另一个可能值。其他情况下，一个值的缺失并不提供任何明显的信息（例如，一个粗心的业务员忘了记录某个客户的数据）。决策树为处理第二种情况提供了一种自然的方法。如果一个实例到达了一个节点，但是由于数据缺失而无法回答该节点对应的问题，人们可以理想化地“将这个实例分成小块”，并且根据所包含训练实例数的比例，将这些小块送往各个分支。如图 6-6 所示，如果 30% 的训练实例往左走，那么一个有缺失值的实例在这个决策节点就被分为两部分，比重 0.3 的那一部分往左走，而比重 0.7 的那一部分往右走。当这个实例不同的小块最终都到达叶子节点时，我们计算对应叶子节点输出值的加权平均，或者计算出一个分布。加权平均中的权重与到达叶子部分的比重成正比。在这个例子中，输出值是 0.3 乘以左边的输出加上 0.7 乘以右边的输出。不用说，以左、右子树作为参数的同一程序的递归调用，是获得非

① 令人好奇的是，在计量经济学领域中广泛应用的一个更通用的版本，称作基尼指数、系数或者比例，被用来描述总体资源配置的不平等性。报纸会定期发布各个国家的基尼指数排名情况，并结合社会经济变量的分析，当然其中的道理不会跟外行解释的。

常紧凑的软件实现的一个方法。

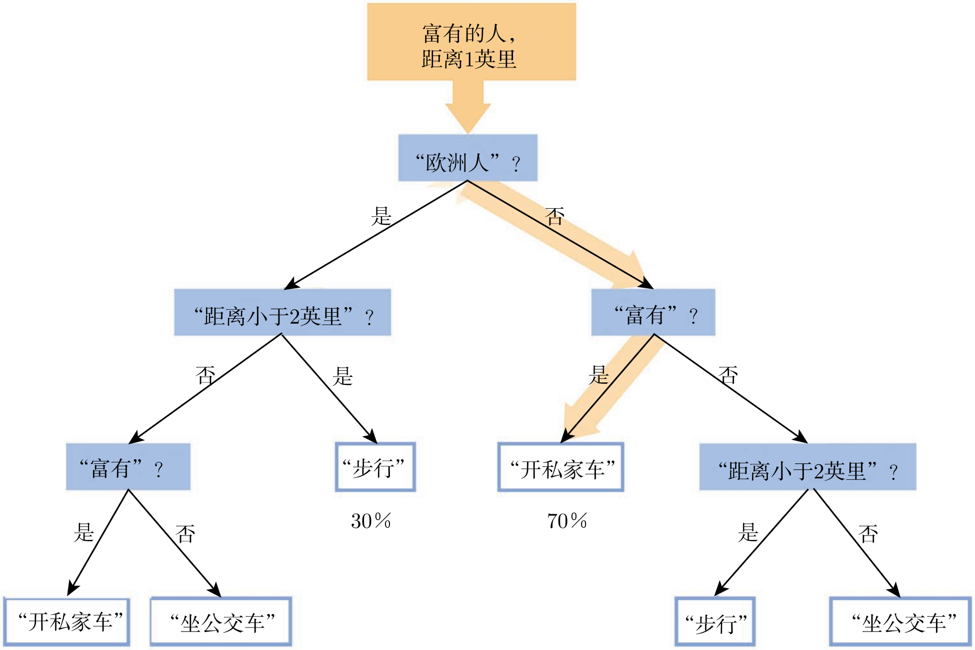


图 6-6　缺失的身份信息。到达顶部节点的数据点被发送到其左侧和右侧的两个子节点，权重根据答案“是”和“否”在训练集中出现的频率而有所不同

最后提醒一下，不要将构建中的决策树（使用已标记实例、纯度度量，以及选择合适的问题）和使用已构建好的决策树相混淆。当使用决策树时，通过一系列决策，输入样本会迅速从树的根节点分配到叶子节点。

6.2　民主与决策森林

在 20 世纪 90 年代，研究人员发现了如何**用民主的方式将学习机集成起来**（例如，比随机猜测性能略好的通用“弱”分类器），以得到更好的准确性和泛化性能 [95, 11]。这可以类比于人类社会：很多情况下，设立**专家团体**（committee of experts）是做出更优质决策的一个方法，专家或者达成共识，或者提出不同的方案并表决（在其他情况下，这也是推迟决策的方法，毕竟所有类比都有它的缺点）。“群众的智慧”[104] 是近来用以强调民主决策积极作用的一个术语。在机器学习中，输出常常由多数决定（对于分类）或由平均决定（对于回归）。

在处理高维数据时，民主式集成这一点似乎尤其正确，因为在现实的应用中，高维数据里可能有很多不相关的属性。这一话题并不像它看起来那样抽象：从光学字符识别中用到的

神经网络 [11] 到游戏机输入设备中树的集成 [39]，已经出现了很多相关的应用 ①。

为了让专家团体来做出高明的决策，这些专家需要有不同的思维方式（即不受群体思维的影响，以同样方式思考的专家是没有用的）并且具有较高的素质。获得不同的树的方式有多种，例如，用不同的实例集来训练它们，或者以不同的随机选择来训练它们（用人来打比方，想想学生会选择同一科目的不同课程）。

- 可以用**自助法**从初始集合中产生**不同的训练实例集**（参见5.3 节）：给定一个大小为 ℓ 的标准训练集 D，自助法通过从 D 中均匀带放回（有些实例可以重复）的抽样产生新的训练集。抽样之后，每个训练集中大概有 $1 - 1/\mathrm{e} \approx 63.2\%$ 的实例是互不相同的，其他的都是重复的。想想随机投球入筐（回想图 5-5）：对于较大的 ℓ，大概只有 63.2% 的筐里有一个或以上的球。筐中的每个球对应一个实例。在每个自助法的训练集中，大概有三分之一的实例被排除了。将自助法用于创造不同样本集的方法称为**装袋法**（bagging，“自助汇合”）：用不同的随机自助样本来构造不同的树，输出是不同树的输出的平均（对于回归问题）或着表决（对于分类问题）。
- 在选择一个节点的最优问题时，可以通过限制选择，在训练时进行**不同的随机选择**。

作为上述划分方法的一个例子，这里来看看它们是如何在**随机决策森林**中实现的 [59, 28]：

- 每棵树用原始数据集的一个自助抽样（即允许重复）来进行训练；
- 每当一个叶子节点必须被分裂的时候，只考虑属性随机选择的一个子集。在极端情况下，只考虑一个随机属性（一维）。

更准确地说，令 d 为输入变量的总数，每棵树是这样构造的：选择 d' 个输入变量用以确定一个节点上的决策，d' 是个较小的数，通常比 d 要小得多（在极端情况下就是 1）。一个自助抽样（“包”）引导一棵树的构造，而那些不在包里的实例可以用来估计树的误差。对于树上的每个节点，d' 个属性是随机选择的，它们是在这个节点上做出决策的基础。我们计算基于这 d' 个变量的最优分割（“最优”要根据所选的纯度标准而定，IG 或者 GI）。每次选择一个类型变量来对一个节点进行分割，可以随机选择这些类型的子集，并定义一个替换变量，当类型值在这个子集中就为 1，否则为 0。每棵树都完全成长而不修剪（就像构造一棵普通的分类树那样）。

通过上述步骤，我们实际上已经组建了一个团体（“森林”），其中的每一个专家（“树”）都已经接受了不同的训练，因为他们已经看到了一组不同的实例（包），也因为他们用不同的观点看待问题（每一个节点使用不同的随机选择的标准）。然而没有哪位专家可以完全保证胜任工作：每个专家关注变量的顺序远远没有达到贪心的标准，因此信息量最大的问题并没有最受关注，如此一来，单独的一棵树是非常弱的；然而，大多数专家都比随机分类器要好，因此多数占优原则（或者加权平均）将会提供合理的答案。

使用自助法时的泛化估计可以在训练过程中以一种自然的方式得到：记录**包外的误差**（不在包中的实例上的误差），并在整片决策森林上求平均。

① 决策森林在微软 Xbox 游戏机主机的 Kinect 传感器中用于人体跟踪。

不同变量（**特征或属性排序**）的相关性也可以在决策森林中用一种简单方式来估计。主体思路是：如果一个类别特征是重要的，那么随机地置换其值应该会导致其性能显著降低。用决策森林拟合数据之后，为了推导第 i 个特征的重要性，将第 i 个特征的值进行随机置换，并重新计算这个扰乱的数据集上的包外误差。求得扰乱前后的包外误差之差，并在所有树上求平均值。误判率增加的百分比与所有变量不变时包外误差率的比值就是该特征所得的分数。误判率增加较多的特征比误判率增加较少的特征更重要。

可以使用大量树（数以千计并不罕见）这一事实意味着，对于需要进行分类或预测的每个实例，会有非常多的可用的输出值。通过收集和分析如此多树的输出的分布，可以得出回归的置信界或者分类的概率。例如，如果有 300 棵树预测“晴天”，剩下的 700 棵树预测“下雨”，那么可以说估计“下雨”的概率是 70%。



梗　概

简单的“如果–那么”规则提炼出在某种程度上可以被人们理解的信息金块。避免可能的规则矛盾所带来的混乱，有一个简单方法是以层次结构来处理问题（首先是信息量最大的），由此引出带组织结构的简单的连续问题，称为**决策树**。

树可以用贪心和递归的方式习得：从一整套的实例集开始，选择一个测试，将它分为两个尽可能纯的子集，再重复产生子集。当子集的纯度足以在树叶上得到分类输出值时，递归过程终止。

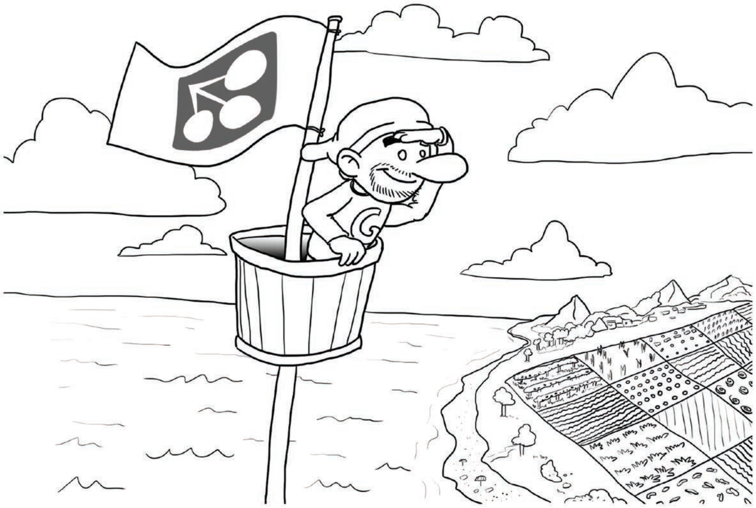
充足的内存和强大的计算能力允许我们训练大量不同的树。通过收集所有输出以及平均（对于回归）或投票（对于分类），它们可以卓有成效地用作**决策森林**。决策森林有各种优点：像所有树那样，它们能自然地处理两类以上的分类问题以及缺失的属性；能提供基于概率的输出，以及概率和误差线；不会有过度训练的风险，因此能很好地泛化到从未见过的数据；由于其并行性，以及每个数据点上减少的测试问题集，它快速而高效。

虽然一棵树的树荫很小，但即使是最火热的机器学习应用，数以百计的树也可以带来清凉。

第 7 章　特征排序及选择

我不介意我的眉毛。虽然我不会说它们是我最好的特征，但它们为我　　增添了些什么。人们告诉我他们喜欢我的眼睛，他们的注意力不在我的眉毛上。

—— 尼古拉斯　霍尔特



从实例中学习模型之前，必须确保输入数据（输入**属性或特征**）足以预测输出。模型建立之后，人们可能愿意**了解**是哪些属性显著影响着输出。如果银行在调查哪些用户足够可靠，可以给他们提供贷款，那么知道哪些因素对信用有正面或负面的影响当然是有意义的。

特征选择，也称为属性选择或变量子集选择，是选择相关特征子集的过程，这些特征将在模型构建中使用。特征选择不同于**特征提取**，特征提取会考虑用原有特征的函数来创建新的特征。

特征选择和排序的问题不是一件小事。假设建立一个线性的模型：

$$y = w_1x_1 + w_2x_2 + \cdots + w_dx_d$$

如果某个权重 w_j 是零，那就很容易推断出对应的特征 x_j 不会影响输出。但要记住，计算机中的数字是不准确的，实例有“噪声”（受测量误差影响），以至于权重为零确实是概率非常小的事件。考虑到非零的权重，可不可以得出这样的结论，即（绝对值）最大的权重都涉及信

息量最大和显著的特征？

可惜不能。这与输入如何“缩放”有关。如果特征 x_j 以千米为单位进行测量时，权重 w_j 很大，那么当这一特征换成以毫米为单位测量时，权重将变得非常小（如果我们希望结果相同，当测量单位改变时，乘积 $w_j \times x_j$ 必须保持恒定）。我们对测量单位的审美变化会立即导致权重改变。特征的值依赖于所选择的单位，因此不能用权重大小来评估其重要性。

尽管如此，如果输入值被归一化，即预乘以某个常数因子，使得典型值的范围相同，例如所有输入变量的大致值域是 0~1，那么线性模型的权重可以给出一些健壮的信息。如果特征选择对于线性模型已经足够复杂了，那么对于非线性模型就更为复杂了。

7.1　特征选择：情境

现在，让我们看一下分类任务的一些定义（见第 3 章的图 3-1），其中输出变量 c 是 N_c 个类之一，输入变量 $\boldsymbol{x}$ 的可能值是一个有限集合。例如，想想预测一个蘑菇究竟是可以食用的（类 1）还是有毒的（类 0）。在数据中提取的所有可能特征集中，人们希望获得信息量高的那些特征，使得分类问题能从足够的信息开始，并且只有分类器的实际结构被保留下来。

现在，你可能会问，为什么人们不使用整套的输入，而只用特征的一个子集。毕竟如果去除一些输入数据，我们也会丢失一些信息。没错，但这里有个**维度诅咒**：如果输入的维数过大，学习任务将变得难以管理。想想在非常高维的空间中用样本来估计概率分布的难度。这是“大数据”文本和网页挖掘应用的标准情况，其中每个文档可以通过成千上万个维度（单词表中每个可能的单词占一个维度）来表征，这样对应于该文档的向量在向量空间中可能是非常稀疏的。

从启发性角度来说，人们的目标是得到**一个小的特征子集，尽可能接近最小的那个，它既包含足以预测输出的信息，又消除了冗余**。这种方式不仅减少了存储器使用量，而且因为消除了不相关的特征和参数，所以泛化性能可以得到改善。此外，人类更容易理解较小的模型。

想想识别手写的文本中的数字。如果这些文本写在彩纸上，并将纸张的颜色作为特征，那么用不同颜色的纸张测试该系统时，学习任务会更加困难，泛化能力也会变差。

特征选择是一个这样的问题：它有许多可能的解决方案，但没有形式上保证的最优解，也不存在什么简单的算法。

首先，应该应用**设计人员的直觉和现有知识**。例如，如果要识别手写的数字，图像应进行缩放和归一化（“五”仍然是五，即使放大、缩小、拉伸、调整亮度，等等），并应该从一开始就去掉明显无关的特征，比如颜色。

其次，需要一种方法来**估计各个特征的相关性或识别能力**，然后可以通过自底向上或自顶向下的方式进行处理，在某些情况下通过重复运行训练模型直接检测待定特征集。一个特征的值与模型的构建方法相关，以及依方法而定的一些评价技术。目前确定了 3 类方法。

- **包装方法**（wrapper method）是“围绕”着特定的预测模型建立的。每个特征子集用来训练一个模型。训练得到的模型的泛化性能可以为该子集评分。包装方法是计算密集型的，但通常为特定模型提供表现最佳的特征集。
- **过滤方法**（filter method）使用代理度量而不是错误率为特征子集评分。常用的度量包括互信息和相关系数。许多过滤器提供特征的排名，而不是一个明确的最佳特征子集。
- **嵌入方法**（embedd method）将特征选择作为模型构建过程的一部分。这种方法的一个例子是用于构建线性模型的 LASSO 方法，它带有回归系数的惩罚，使得其中许多系数收缩到零，从而相应的特征可以消除。另一种方法是递归特征消除，常与支持向量机一起使用，反复构建一个模型，并删除低权重的特征。

通过将过滤方法与包装方法相结合，人们可以用自底向上或自顶向下的方式进行处理。在一个**自底向上的贪心式包含方法**中，人们根据单个特征的识别能力的顺序来逐步添加特征，并通过验证组输出误差是否减少来检验效用。特征的最优数量可以用启发式的方法确定，即验证集上测量的输出误差停止下降时的数量。实际上，如果超过该数量点时仍添加更多的特征，误差可能保持稳定，甚至因为过拟合而逐渐增加。

在**自顶向下的截断法**中，人们从完整的特征集开始，逐步消除特征，同时寻找最佳性能点（持续检查在一个合适的验证集上的误差）。

使用过滤方法必须谨慎。注意，**对单独特征分别进行测量将抛弃它们之间的相互关系**，因此结果只是近似。还有可能发生的是，没有关联信息的两个单独特征将被丢弃，即使它们的组合将会完美地预测输出，试想一个带两个输入的异或函数。

作为异或的一个例子，假设需要识别的类是CorrectMenu(*汉堡包*, *甜点*)，其中两个变量*汉堡包*和*甜点*如果在菜单上，那么对应值 1（如果存在），否则为 0（见图 7-1）。为了在快餐中

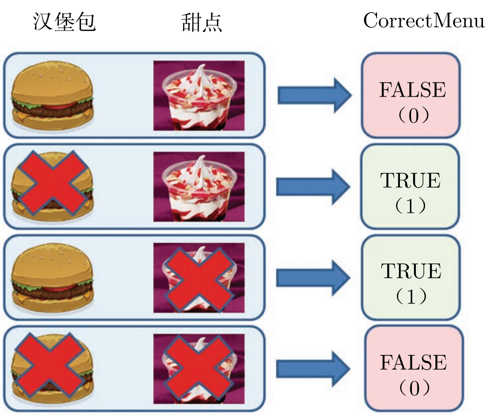


图 7-1　有两个二元输入和一个输出的分类器。单个特征分开来看都不具有信息量，对于一个正确的输出，这两个特征的结合是充分必要的

获得适度的热量，你需要吃一个汉堡包或甜点，但不能同时吃这两个。菜单中的汉堡包（或甜点）单独存在或不存在不能反映菜单是否正确分类。但仅因为它们的单独信息与输出分类无关就消除其中一个或两个输入是不明智的。你需要保存和读取这两个属性来为你的饮食正确分类！这个简单的例子可以泛化：任何饮食专家都会告诉你，营养不是单独某种食物的多少，而是整体组合的平衡。

现在已经明确了情境，接下来考虑**单个特征的识别能力的代理度量**的一些例子。

7.2　相关系数

设 Y 是与输出分类相关联的随机变量，$\Pr(y)$（$y \in Y$）表示输出是 y 的可能性；X_i 是与输入变量 x_i 相关联的随机变量，X 是输入向量的随机变量，它的值是 x。

数值变量间的线性关系使用最广泛的度量是皮尔逊积矩**相关系数**（correlation coefficient），它是通过将两个变量的协方差除以它们的标准差的乘积得到的。采用上述符号，第 i 个输入特征 X_i 和分类的结果 Y 之间，关于期望值 μ_{X_i} 和 μ_Y 以及标准差 σ_{X_i} 和 σ_Y 的相关系数 $\rho_{X_i,Y}$ 定义为：

$$\rho_{X_i,Y} = \frac{\mathrm{cov}[X_i, Y]}{\sigma_{X_i}\sigma_Y} = \frac{E[(X_i - \mu_{X_i})(Y - \mu_Y)]}{\sigma_{X_i}\sigma_Y} \tag{7.1}$$

其中 E 是变量的期望值，cov 是协方差。经过简单的变换，可以得到等价的公式：

$$\rho_{X_i,Y} = \frac{E[X_iY] - E[X_i]E[Y]}{\sqrt{E[X_i^2] - E^2[X_i]}\ \sqrt{E[Y^2] - E^2[Y]}} \tag{7.2}$$

相关系数除以标准差，使其与测量单位相独立（例如，以千米或毫米为单位进行测量，会产生相同的结果）。相关系数的取值是从 -1 到 1。相关性接近 1 意味着正向线性关系（特征值 x_i 相对均值所产生增量通常伴随着结果 y 的增加），接近 -1 则表示反向线性关系。系数越接近零，变量之间的相关性就越弱，例如 (x_i, y) 点图看起来像一片围绕着预期值的各向同性的云，没有明显的方向，如图 7-2 所示。

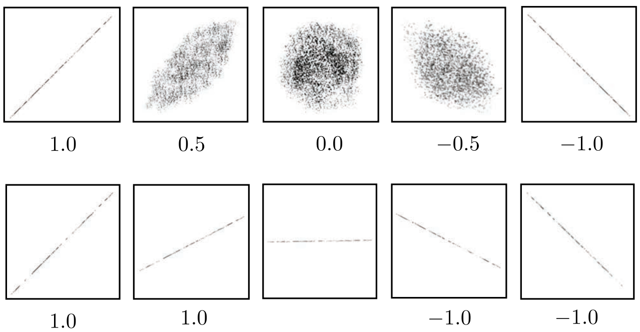


图 7-2　数据分布和对应的相关系数值的例子。记住，相关值都除以标准差，因此下面一排线性分布都有相同的最大相关系数（1 或 −1）

正如之前提到的，统计独立的变量具有零相关性，但零相关并不意味着变量是独立的。相关系数只检测两个变量之间的线性相关性：一个变量可能有第二个变量的充分信息，甚至能直接确定第二个变量的值，如 $y = f(x_i)$ 的情况，但它们还是零相关的。

通常的建议是，不要盲目使用排序标准，无论是上述的还是其他的，要得到（验证）数据上的分类性能测试的实验结果的支持才能使用，正如在包装方法中所做的。

7.3 相关比

很多情况下，学习算法的预期结果是类别型（回答“是/否”或一组有限的选择）。使用相关系数需要假定输出是数值型，因此对于类别型不适用。为了找出一般的相关关系，可以使用相关比（correlation ratio）的方法，衡量数值型输入和类别型输出之间的关系。

相关比背后的基本思想是，根据所观察到的结果将样本的特征向量划分成类。如果一个特征是显著的，那么它应该可以确定至少一个结果类，这个类中的该特征的平均值与其他所有类的平均值是明显不同的，否则该特征对于分辨结果将不太有用。

假设有 ℓ 个样本特征向量，可能是之前阶段试图测量时使用的算法收集到的。令 ℓ_y 表示结果 $y \in Y$ 出现的次数，这样就可以通过结果划分样本特征向量：

$$\forall y \in Y \qquad S_y = \left((x_{jy}^{(1)}, \cdots, x_{jy}^{(n)}); j = 1, \cdots, \ell_y\right)$$

换句话说，元素 $x_{jy}^{(i)}$ 是 ℓ_y 个结果为 y 的样本中的第 j 个样本向量的第 i 个分量（特征）。让我们关注所有样本向量中的第 i 个特征，并计算它在每个结果类中的平均值：

$$\forall y \in Y \qquad \bar{x}_y^{(i)} = \frac{1}{\ell_y} \sum_{j=1}^{\ell_y} x_{jy}^{(i)}$$

和整体的平均值：

$$\bar{x}^{(i)} = \frac{1}{\ell} \sum_{y \in Y} \sum_{j=1}^{\ell_y} x_{jy}^{(i)} = \frac{1}{\ell} \sum_{y \in Y} \ell_y \bar{x}_y^{(i)}$$

最后，特征向量的第 i 个分量与结果之间的**相关比**由下式给出：

$$\eta_{X_i,Y}^2 = \frac{\displaystyle\sum_{y \in Y} \ell_y (\bar{x}_y^{(i)} - \bar{x}^{(i)})^2}{\displaystyle\sum_{y \in Y} \sum_{j=1}^{\ell_y} (x_{jy}^{(i)} - \bar{x}^{(i)})^2}$$

如果第 i 个特征分量与结果的值之间的关系是线性的，那么相关系数和相关比都等于依赖性的斜率：

$$\eta_{X_i,Y}^2 = \rho_{X_i,C}^2$$

在所有其他情况下，相关比可以把握非线性依赖。

7.4 卡方检验拒绝统计独立性

让我们再来考虑二分类问题与一个单独的二元特征。例如，在文本挖掘中，特征可以表示文档中一个特定词语（关键字）t 的存在/不存在和输出，输出可以指示该文档是不是关于编程语言的。因此，我们在做的是评估**两个类别型特征之间的关系**。

人们可以从导出 4 个计数器 $\text{count}_{c,t}$ 开始，它们对属于（或不属于）给定类的实例文档中有（或没有）的词语 t 进行计数。例如，$\text{count}_{0,1}$ 对应类 =0 且有词语 t，$\text{count}_{0,0}$ 对应类 =0 且没有词语 t　　然后，就可以通过将计数除以实例总数 n 来估计概率。

零假设是"存在词语 t"和"文档属于类 c"这两个事件是独立的。在此假设下，联合事件的上述计数的预期值可以通过单个事件的概率的相乘获得。例如 $E(\text{count}_{0,1}) = n \cdot \Pr(类 = 0) \cdot \Pr(存在词语\ t)$。

如果计数偏离两个独立事件的期望值，人们可以得出这两个事件相关的结论，因此该特征对于预测输出是显著的。接下来只需检查*偏差是否足够大，以保证它不是偶然发生的*。一个统计上合理的测试是**统计假设检验**。

统计假设检验是通过使用实验数据做出统计决定的方法。在统计学中，**如果某个结果不太可能是偶然发生的，该结果称为统计学显著的**。"显著性检验"这一短语是由罗纳德　费希尔在 1925 年左右提出的，他是奠定了现代统计科学基础的天才。

假设检验有时被称为**验证性数据分析**，与探索性数据分析相对。决策几乎都是用零假设检验做出的，也就是必须回答这样的问题：假定零假设为真，观察到一个至少与实际观察到的值一样极端的检验统计值的概率是多少？

在我们的例子中，测量 χ^2（卡方）值：

$$\chi^2 = \sum_{c,t} \frac{\left[\text{count}_{c,t} - n \cdot \Pr(类 = c) \cdot \Pr(词语 = t)\right]^2}{n \cdot \Pr(类 = c) \cdot \Pr(词语 = t)} \tag{7.3}$$

χ^2 值越高，观察到的数据支持独立性假设的信念就越小。如果希望得到定量值，可以通过标准统计公式计算一个特定值偶然发生的概率。

对于特征排序而言，没有必要进行额外的计算，粗略的值也够用了：根据这一标准，最佳特征是具有较高 χ^2 值的。它们更加偏离独立的假设，因此可能是相关的。

7.5 熵和互信息

"有信息量的特征"这一定性标准，可以用统计方式和**互信息**（MI）的概念进行精确的定义。

输出分布的*不确定性*的特点可以用输出的概率分布进行测量。理论上合理的测量不确定性的方式是使用**熵**（entropy），参见下面的详细定义。现在，我们知道一个特定的输入值 x，

输出的不确定性会随之降低。输入某个值之后，输出中不确定性减小的量称为**互信息**。

如果一个特征和输出之间的互信息为 0，输入的知识并不会减少输出中的不确定性。换句话说，不能（单独地）使用所选择的特征以预测输出 —— 无论我们的模型有多么先进。因此，输入特征向量和输出（期望的预测）之间的 MI 度量与确定有希望的（有信息量的）的特征是非常相关的。参考文献 [6] 开创性地使用互信息来进行特征选择。

在信息论中，**熵**即输出类（随机变量）的统计不确定性的测量，定义为：

$$H(Y) = -\sum_{y \in Y} \Pr(y) \log \Pr(y) \tag{7.4}$$

熵量化平均信息，单位是二进制位（bit），用于指定哪个事件发生（见图 6-5）。它也可以用来量化在不丢失信息的情况下，一个消息可以被压缩的程度 ①。

现在来计算第 i 个输入特征 x_i 对分类的结果 y 的影响。知道输入特征值（$X_i = x_i$）后，Y 的熵是：

$$H(Y|x_i) = -\sum_{y \in Y} \Pr(y|x_i) \log \Pr(y|x_i)$$

其中，$\Pr(y|x_i)$ 给定第 i 个特征的值为 x_i 且类型为 y 的条件概率值。

最后，变量 Y 的条件熵（conditional entropy）被定义为 $H(Y|x_i)$ 在第 i 个特征所能取的所有值 $x_i \in X_i$ 上的期望值：

$$H(Y|X_i) = E_{x_i \in X_i}\left[H(Y|x_i)\right] = \sum_{x_i \in X_i} \Pr(x_i) H(Y|x_i) \tag{7.5}$$

条件熵 $H(Y|X_i)$ 总是小于或等于 $H(Y)$。它等于 $H(Y)$ 当且仅当第 i 个输入特征和输出类统计上是独立的，即对于每个 $y \in Y$ 和 $x_i \in X_i$，联合概率 $\Pr(y, x_i)$ 都等于 $\Pr(y)\Pr(x_i)$（注：这个定义并没有谈论线性相关性）。根据定义，不确定性减小的量就是变量 X_i 和 Y 之间的互信息：

$$I(X_i; Y) = I(Y; X_i) = H(Y) - H(Y|X_i) \tag{7.6}$$

使得 X_i 和 Y 之间的对称性明显的一个等效表达式是：

$$I(X_i; Y) = \sum_{y, x_i} \Pr(y, x_i) \log \frac{\Pr(y, x_i)}{\Pr(y)\Pr(x_i)} \tag{7.7}$$

虽然理论上很强大，但从已标记的样本开始，估计高维特征向量的互信息，这不是一个简单的任务。参考文献 [6] 中提出了一个只使用单个特征和输出之间的互信息的启发式方法。

① 香农的信源编码定理表明，在极限的情况下，消息的二进制字母表可行最短编码的平均长度是它们的熵。如果事件的发生概率相同，那么是不可能进行压缩的。如果概率不同，最可能的事件可以分配较短的码，由此压缩了信息整体的长度。这就是为什么 zip 工具可以成功地压缩有意义的文本，这样的文本中单词和短语有不同的出现频率，但压缩伪随机序列却有困难，比如 JPEG 文件或其他图像文件的有效编码。

需要强调的是，互信息与相关性不同。一个特征对于输出可以是很有信息量的，即使二者不是线性相关的，而互信息度量甚至不要求两个变量是数值型的。请记住，名义类变量具有两个或更多类别，但这些类别没有内部的排序。例如，性别是分为两类（男性和女性）并且没有内在排序的标称变量。如果你有足够丰富的数据来估计它，那么互信息应该是最好的衡量信息含量的办法。



梗 概

减少模型所使用的输入特征的数量，同时又能保持大致相同的性能，这样做有许多优点：更小的模型和更高的可理解性，更快的训练和更短的运行时间，可能还有更强的泛化能力。

对特征进行排序，如果不考虑特定的建模方法以及它们之间的相互关系，将会是很困难的。想想一个侦探（在这种情况下，分类的目标是“有罪”或“无罪”）聪明地结合多个线索，并避免混乱的论证。排序和过滤只是试探的第一步，并且需要通过所选的方法尝试不同的特征集进行验证，将方法用特征选择方案“包装”起来。

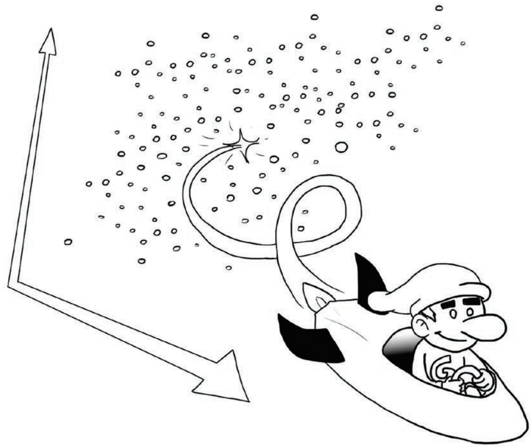
一个简便的方法是：仅当有理由猜测是线性关系时，才信任**相关**系数，否则可以考虑其他相关度量，尤其是相关比，即使输出值不是定量的也适用。使用**卡方**来确认输入和输出之间可能的依赖性，通过估计单独和联合事件的概率。最后，可以利用强大的**互信息**来估计定性或定量特征之间的任意依赖关系，但要注意，只有非常少的几个实例时，结果可能会高估。

作为一个练习，挑选你自己喜欢的福尔摩斯的故事，并找出他使用了哪些特征（线索、证据）选择方法来揭露和逮捕真凶，并让他的朋友华生叹服。

第 8 章　特定非线性模型

想学会飞翔，就必须先学会站立、行走、奔跑、攀登和跳舞，没有人能直接飞翔。

—— 尼采



本章继续沿着从线性模型到非线性模型的道路进行探索。为了避免突兀，我们先不推出最一般化和强大的模型，而是先从线性模型的逐渐修改开始，先使其适用于预测概率（**logistic 回归**），然后使线性模型*局部化*，更加关注最接近的实例，就像一种平滑的 K 近邻法（**局部加权线性回归**），最后通过对权重的适当限制选择输入子集（LASSO）。

准备阶段之后，接下来的章节中将接触到灵活非线性模型的精髓，即任意平滑输入–输出关系，例如多层感知器（Multi-Layer Perceptron，MLP）和支持向量机（Support Vector Machine，SVM）。

8.1　logistic 回归

在统计学中，logistic 回归被用于根据一组历史事件的记录**预测分类变量的各种结果的概率**。例如，从可能患心脏疾病的病人的相关数据入手（疾病“有”或者“没有”是分类输出变量），想要预测一个新来的病人患心脏疾病的概率。这个名字有一定程度上的误导性，事实上它是一种分类技术，而不是回归。但这种分类是通过概率的估计得到的，因此使用术语“回

归”。常见的输出是二元的，也就是有两个可用类别。

使用线性模型的问题是，输出值是无界的，而我们需要限定输出值范围为 0~1。logistic 回归大部分是由一个线性模型运行的，但 **logistic 函数**（见图 8-1）被用来转化线性预测器的输出，从而取得一个 0~1 的值，这也可以解释为概率。通过将这一概率值与阈值相比较，可以实现一种分类（例如，当输出概率大于 0.5 时，分类为“是”）。

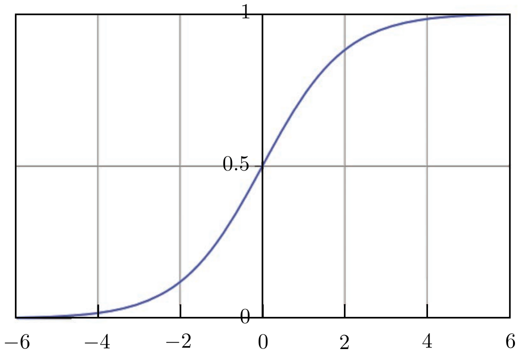


图 8-1　logistic 函数把输入值以平稳的方式转换为 0~1 的输出值。该函数的输出可以被阐释为一种概率

logistic 曲线是一种常见的 S 型函数。logistic 一词是这一函数用于研究人口增长时引入的。在人群中，繁殖率正比于现有的人口数量和可用资源的数量。当人口增长时，可用的资源减少；当人口达到系统的承载能力时，资源数量为零。增长的初始阶段接近指数关系；然后进入饱和阶段，增长放缓；到成熟阶段时，增长停止。

标准的 logistic 函数由下式定义：

$$P(t) = \frac{1}{1+\mathrm{e}^{-t}}$$

其中 e 是欧拉数（数学常数）。变量 t 可以是时间，不过这里 t 是线性模型的输出，想想式 (4.1)，我们有：

$$P(\boldsymbol{x}) = \frac{1}{1+\mathrm{e}^{-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x})}}$$

记住，线性模型 $\boldsymbol{w}$ 也可以包括一个常数值 w_0，前提是在输入值列表中人为添加一个总是等于 1 的输入值 x_0。

下面来看看在这种情况下最大化的是哪个函数。线性变换权重的最佳值是通过**最大似然估计**来确定的，即通过最大化得到这些输出值的概率，这些输出值事实上是从给定的已标记实例中得到的。将每个独立事件的概率相乘。令 y_i 为所观察到的输出（1 或 0），对应输入为 $\boldsymbol{x}_i$。若 $\Pr(y=1|\boldsymbol{x}_i)$ 是由该模型得到的概率，正确分类为 1，则获得测量输出值 y_i 的概率为 $\Pr(y=1|\boldsymbol{x}_i)$；而若正确分类标签为 0，则 $\Pr(y=0|\boldsymbol{x}_i)=1-\Pr(y=1|\boldsymbol{x}_i)$。所有因子需要相乘

得到所有实例的整体概率。按照惯例使用对数形式，于是因子（每个实例对应一个）求积转换成了求和：

$$\text{LogLikelihood}(\boldsymbol{w}) = \sum_{i=1}^{\ell}\Big\{y_i \ln \Pr(y_i|\boldsymbol{x}_i, \boldsymbol{w}) + (1-y_i)\ln(1-\Pr(y_i|\boldsymbol{x}_i, \boldsymbol{w}))\Big\}$$

似然率 Pr 与系数（权重）$\boldsymbol{w}$ 的相关性已经非常明确了。

由于上述表达式中的非线性性，我们不可能找到使似然函数最大化的权重的解析表达式，而必须用迭代过程来代替，例如梯度下降法。该过程始于一个初始解决方案 $\boldsymbol{w}_{\text{start}}$，然后通过向负梯度的方向移动来进行细微调整，观察是否需要改进，并重复这一步骤直到改进十分微细，此时我们认为这一过程已经收敛。

像往常一样，在 ML 中，人们关心怎样使泛化最大化。当在验证集上测量并估计出的泛化性能为最佳的时候，以上最小化过程可以也应该尽早被终止。

8.2 局部加权回归

4.1 节中我们已经看到如何确定一个线性关系的系数。第 2 章中的 K 近邻法根据最接近旧的（已标记的）实例的输出值来预测新输入的输出值，给出的输出可以是最接近已储存的一个输入所对应的输出值的那个，或者是选出的最近邻值的输出值的某些简单组合。

本节中考虑的方法类似于最近邻值的输出的线性组合。但我们没有那么残酷，不会只关注 K 个最近邻值而消除所有其他值的影响。这是一种平滑的变化：我们**根据和被预测的实例之间的距离来逐渐减少实例对预测的影响**，而不是选择一组 K 个胜者。

通过加权得到的整体相关性可能会相当复杂。当模型需要在不同的点进行评估时，线性回归仍然可以使用，只不过该评价点附近的评估点被认为比远处的“更重要”。这里遇到了一个非常普遍的原则：在（自然的或自动的）学习中，相似的实例通常被认为比那些相差甚远的更相关。

局部加权回归（Locally Weighted Regression）是一种*懒惰的*基于存储的技术，这意味着所有点和评估值都被存储了，而只有查询特定的某点的时候才会*基于请求*建立特定的模型。

为了预测一个点 $\boldsymbol{q}$（称为*查询点*）的评估结果，我们对训练点应用线性回归。为了确保在确定回归参数过程中的局部性（相近的点更相关），给每个样本点分配一个*权重*，这个权重会随着与查询点距离的增加而减小。值得注意的是，在神经网络业内，术语“权重”一般情况下指由训练算法计算得到的模型参数，然而在这种情况下，权重度量每个训练样本的重要性。为了避免混淆，我们用术语*重要性*和符号 s_i（下面所用的对角矩阵记为 $\boldsymbol{S}$）来表示这一特定用法。

如 4.1 节所述，我们假设所有输入向量 $\boldsymbol{x}_i$ 都以常数 1 作为第 0 个元素，它被用作回归中的常数项，因此全部等式的维数实际上是 $d+1$。

加权后的最小二乘拟合的目标是最小化下面的加权误差（式 (4.2) 中隐式地假设了每个点的权重是一样的）：

$$\mathrm{error}(\boldsymbol{w}; s_1, \cdots, s_n) = \sum_{i=1}^{\ell} s_i(\boldsymbol{w}^{\mathrm{T}} \cdot \boldsymbol{x}_i - y_i)^2 \tag{8.1}$$

从 4.1 节中用弹簧进行类比的观点来看，样本点不同的权重分布对应于使用不同弹性常数（强度）的弹簧，如图 8-2 所示。为了最小化式 (8.1)，可以令其关于 $\boldsymbol{w}$ 的梯度等于 0，得到如下解：

$$\boldsymbol{w}^* = (X^{\mathrm{T}}\boldsymbol{S}^2X)^{-1}X^{\mathrm{T}}\boldsymbol{S}^2\boldsymbol{y} \tag{8.2}$$

其中 $\boldsymbol{S} = \mathrm{diag}(s_1, \cdots, s_d)$，而 X 和 $\boldsymbol{y}$ 则是根据式 (4.5) 来定义。注意当所有权重相等时，式 (8.2) 简化为式 (4.5)。

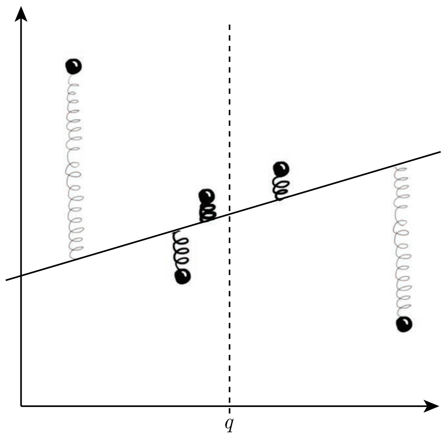


图 8-2 加权最小二乘拟合的弹簧类别（与图 4-6 比较）。现在，弹簧有不同的弹性常数，较粗意味着较硬，所以它们对整体势能的影响需要进行加权。对于上述情况，较硬的弹簧代表更靠近查询点 $\boldsymbol{q}$ 的那些点

根据储存样本到查询点的距离，可以使用以下函数来描述它们的重要性：

$$s_i = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{q}\|^2}{W_K}\right)$$

其中 W_K 是度量“核宽度”的一个参数，即对远距离实例的灵敏度；当距离远大于 W_K 时，重要性迅速衰减至 0。

图 8-3（上）给出了一个例子，模型需要在查询点 $\boldsymbol{q}$ 进行估值。样本点 x_i 用圆圈来表示，它们的重要性 s_i 随着与 $\boldsymbol{q}$ 点距离增加而减小，并且用内部阴影深浅程度来表示，黑色意味着重要性最高。线性拟合（实线）是通过考虑各点的重要性而计算出的，并根据模型在 $\boldsymbol{q}$ 点估计出相应的值。对于每一个查询点，每个样本点的重要性和随后的线性拟合都会重新计算，前提是曲线如图 8-3（下）所示。

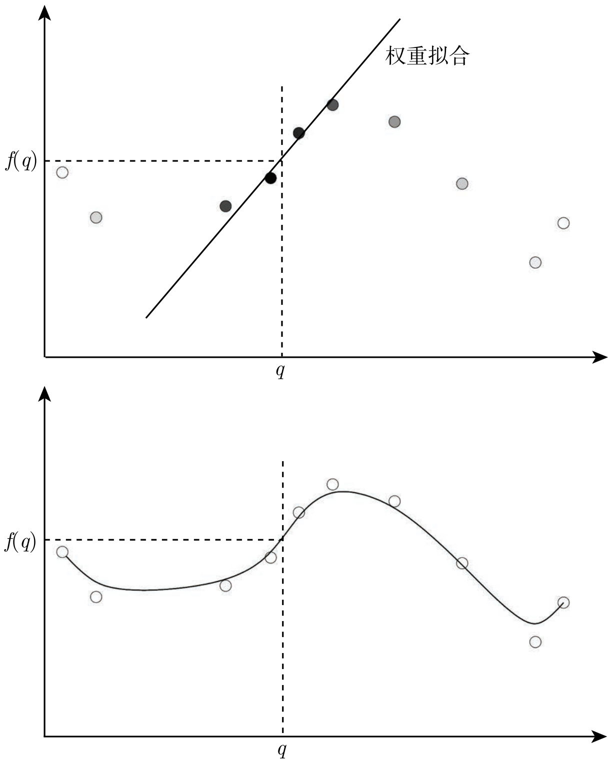


图 8-3 （上）LWR 模型在 $\boldsymbol{q}$ 点估值，样本点的重要性由内部阴影表示；（下）所有点的估值，每个点对应一个不同的线性拟合

贝叶斯局部加权回归

目前为止，我们还没有做过有关待定系数的先验概率分布的假设。在某些情况下，一些与任务有关的信息可以方便地通过先验概率分布加入模型。

贝叶斯局部加权回归，记为 B-LWR，用于确定关于系数取值的先验信息。贝叶斯技术的力量一般来自明确规范的模型假设和参数（比如，一个**先验分布**可以建模我们对函数的初步认识），而且建模的概率并不局限于它的期望值，而是整个概率分布。例如，**置信区间**可以定量地得出期望值的不确定性。

为我们带来 B-LWR 的是系数 w 分布的先验假设：它们服从零期望和协方差矩阵为 $\boldsymbol{\Sigma}$ 的多元高斯分布。而 σ 的先验假设是 $1/\sigma^2$ 服从以 k 为形状参数、以 θ 为尺度参数的伽马分布。由于使用加权回归，每个点及其相应输出是通过高斯权重函数加权的。以矩阵的形式，数据点的权重组成 $\ell \times \ell$ 的对角矩阵 $\boldsymbol{S} = \mathrm{diag}(s_1, \cdots, s_\ell)$，而矩阵 $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \cdots, \sigma_\ell)$ 包含 $\boldsymbol{w}$ 分布的先验方差。

查询点 $\boldsymbol{q}$ 的局部模型是由 $\boldsymbol{w}$ 的边缘后验分布预测的，而 $\boldsymbol{w}$ 的期望值由下面的式子估计：

$$\bar{\boldsymbol{w}} = (\boldsymbol{\Sigma}^{-1} + X^T\boldsymbol{S}^2X)^{-1}(X^T\boldsymbol{S}^2\boldsymbol{y}) \tag{8.3}$$

注意，去除先验知识其实对应先验假设中的方差无穷大，从而 $\boldsymbol{\Sigma}^{-1}$ 变成零矩阵，式 (8.3) 简化为式 (8.2)。矩阵 $\boldsymbol{\Sigma}^{-1} + X^{\mathrm{T}}\boldsymbol{S}^2X$ 是加权协方差矩阵，其中以 $\boldsymbol{w}$ 的先验知识加以补充。它的逆矩阵表示为 $\boldsymbol{V_w}$。基于 ℓ 个数据点的高斯噪声的方差可以估计为：

$$\sigma^2 = \frac{2\theta + (\boldsymbol{y}^{\mathrm{T}} - \boldsymbol{w}^{\mathrm{T}}X^{\mathrm{T}})\boldsymbol{S}^2\boldsymbol{y}}{2k + \sum_{i=1}^{\ell} s_i^2}$$

$\boldsymbol{w}$ 分布的协方差矩阵的估计值由下面的式子计算：

$$\sigma^2\boldsymbol{V_w} = \frac{(2\theta + (\boldsymbol{y}^{\mathrm{T}} - \boldsymbol{w}^{\mathrm{T}}X^{\mathrm{T}})\boldsymbol{S}^2\boldsymbol{y})(\boldsymbol{\Sigma}^{-1} + X^{\mathrm{T}}\boldsymbol{S}^2X)}{2k + \sum_{i=1}^{\ell} s_i^2}$$

自由度由 $k + \sum_{i=1}^{\ell} s_i^2$ 给定。从而查询点 $\boldsymbol{q}$ 的预测输出响应为：

$$\hat{y}(\boldsymbol{q}) = \boldsymbol{q}^{\mathrm{T}}\bar{\boldsymbol{w}}$$

而预测输出的均值的方差可由下面的式子计算：

$$\mathrm{var}(\hat{y}(\boldsymbol{q})) = \boldsymbol{q}^{\mathrm{T}}\boldsymbol{V_w}\boldsymbol{q}\sigma^2 \tag{8.4}$$

8.3　用 LASSO 来缩小系数和选择输入值

考虑线性回归模型时，*岭回归*是一种通过二次方式来惩罚大系数，从而使得模型更稳定的方法，如式 (4.7) 所示。

普通的最小二乘估计法通常偏差较小，但是方差较大。为了提高准确率，有时可以将一些系数缩小或者设置为零。通过这样做，我们牺牲一点偏差，以减少预测值的方差，从而可以提高整体的预测准确率。还有一个原因是**便于解释**。如果存在大量的预测量（输入变量），我们通常想找到一个具有最大影响力的较小子集。特征子集选择和岭回归，这两个改进估计的标准技术仍然存在一些缺陷。子集选择提供了便于解释的模型，但由于它是一个离散过程，输入变量（回归量）要么保留，要么删除，该模型的变化也可能是特别大的。即使数据中很小的变化，也可能产生十分迥异的模型，这就降低了预测准确率。岭回归是一种连续让系数缩小，从而使得模型更稳定的过程，然而它并没有设置任何系数为零，所以无法得出一个易于解释的模型。参考文献 [107] 中提到一种新的技术**LASSO**，即“最小绝对收缩和选择算符”。它使得一些系数缩小而另一些设置为零，因此保持了子集选择和岭回归两种方法的优势。

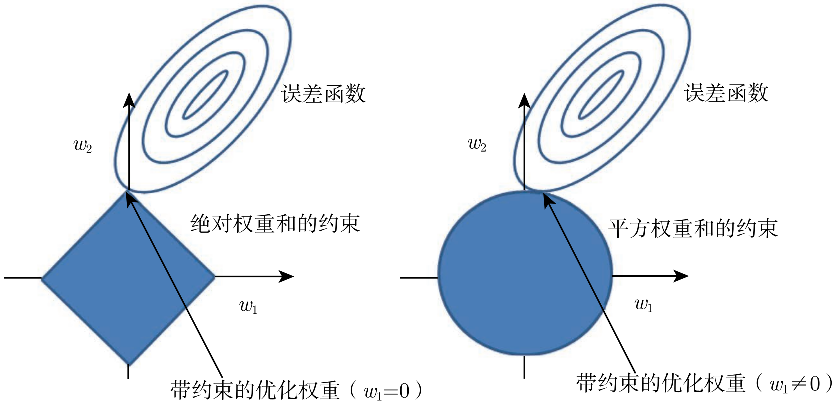


图 8-4 在 LASSO 中，最好的解决方案出现在二次误差函数等高线接触正方形处，有时会在正方形的角上，对应某些零系数。相反，岭回归的二次约束没有角来让等高线接触，因此权重中很少会产生零

LASSO 使用**权重绝对值**的总和作为约束 $\|\boldsymbol{w}\|_1$（参数向量的 L_1 范数），它不会大于给定值。LASSO 在系数绝对值总和小于一个常数的约束下，使得残差平方和最小化。通过一个标准技巧，将带约束的优化问题通过拉格朗日乘数法转化为无约束的问题，这相当于将 $\lambda\|\boldsymbol{w}\|_1$ 加入无约束最小化的最小二乘：

$$\text{LASSOerror}(\boldsymbol{w};\lambda)=\sum_{i=1}^{\ell}(\boldsymbol{w}^{\mathrm{T}}\cdot\boldsymbol{x}_i-y_i)^2+\lambda\sum_{j=0}^{d}|w_j| \tag{8.5}$$

LASSO 和岭回归一个最主要的区别是，在岭回归中随着惩罚的增加，所有系数减小，但保持非零的状态，而 LASSO 随着惩罚的增加会导致更多的系数变为零。对应的权重为零的输入值就可以消除，从而导致模型使用较少的输入值（输入的**稀疏化**），因此更便于解释。较少的非零参数有效减少了变量数目，而这正是影响解决方案的因素。换言之，**作为模型构建过程的一部分**，LASSO 是一种进行**特征选择**的嵌入式方法。

注意，式 (8.5) 中惩罚较大权重的那一项，当权重为零时不存在导数（偏导数从对应负值的 -1 跳到对应正值的 $+1$）。通过计算导数并令其等于零来求解，并得到一个线性系统的“技巧”在这里没法使用。优化 LASSO 的问题可以通过引入带线性不等式约束的**二次规划**或更一般的凸优化方法来解决。LASSO 的参数 λ 的最佳值可以通过交叉验证来获得。

拉格朗日乘数优化约束问题

上述方法将带约束的优化问题转化为无约束的优化问题，并已被广泛应用，对于好奇心旺盛的读者，即使是关于数学的题外话也是值得一提的。在数学优化中，**拉格朗日乘数法**是在带约束的前提下用来寻找函数局部最大值和最小值的方法。带约束问题是通过将各个约束乘以一个参数（一个拉格朗日乘数）转化为无约束的。最小化转换后的函数将导出优化的必

要条件。

考虑一个二维问题：

$$\begin{aligned} \text{最大化} \quad & f(x,y) \\ \text{约束条件} \quad & g(x,y)=c \end{aligned}$$

对于不同的 d，我们可以可视化 f 的等高线

$$f(x,y)=d$$

在图 8-5 中展示了这些等高线和 $g(x,y)=c$ 的等高线。

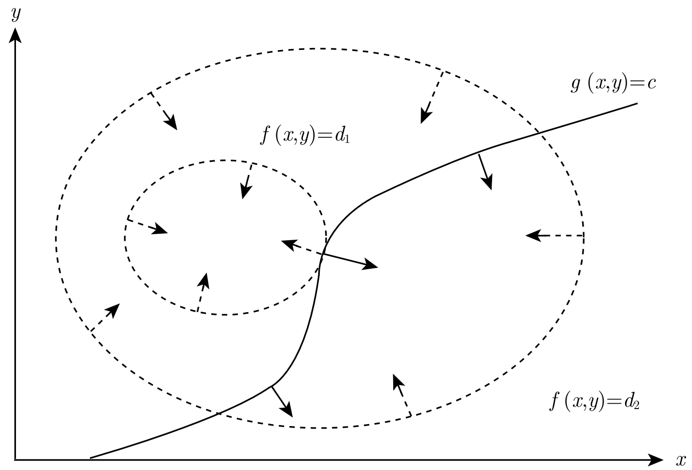


图 8-5 拉格朗日乘数法

假设我们沿着等高线 $g=c$ 走。一般情况下，f 和 g 的等高线是不同的，所以 $g=c$ 的等高线将与 f 的等高线相交或者穿过。这就相当于，沿着 $g=c$ 的等高线移动，f 值是不同的。只有当等高线 $g=c$ 与 f 的等高线相切（接触但不交叉）时，f 的值既不增加也不减少。

当 f 和 g 的**等高线切向量平行**时，f 和 g 的等高线相接触。因为函数的梯度是垂直于等高线的，所以切向量平行就相当于 f 和 g 的梯度是平行的。因此我们在点 (x,y) 处要求 $g(x,y)=c$，并且：

$$\nabla f(x,y)=\lambda\nabla g(x,y)$$

拉格朗日乘数 λ 确定了一个梯度需要乘以什么来获得另一个梯度！

梗 概

线性模型应用广泛，但是在许多情况下仍有不足。本章举例介绍了 3 个具体的改进方法。

首先，有些原因可能导致输出值需要被限定在一定的范围内。比如，若要预测概率，则输出值的范围只能是 0~1。一种方法是将输入的线性组合传递给一种“挤压”logistic 函数。最大化训练事件的对数似然率，就得到了广泛使用的**logistic 回归**。

其次，可能有些情况下线性模型需要*局部化*，不同的输入点赋予不同的权重，那些距离需要预测的输入样本更近的点拥有更大的权重。这就是**局部加权回归**。

最后，在需要优化的函数中加入的大权重的惩罚项，不一定是权重的平方和（只有通过计算导数得到线性等式时才会有），可以有其他选择。例如，用绝对值之和作为惩罚，既可以有成效地减少权重，又能使输入变得稀疏。这就是使用 LASSO 技术来缩小系数和选择输入。LASSO 减少了非零权重的数目，从而也减少了对输出值有影响的输入值数量。

学习本章之前，对于你来说套索（lasso）只是一端有活结的、用来套住牛马的长绳。而现在，你可以用它来套住更多有意义的模型。

第 9 章　神经网络：多层感知器

大自然是大师中的大师，除非从他那里获得灵感，否则其他的都是徒劳无益。

——达　芬奇



人类的**神经系统**，包含大约 1000 亿个计算单元和大约 10^{15} 个连接，能够完成令人惊讶的智能行为。事实上，人类大脑的能力定义了智能。这些计算单元是称为**神经元**的一类特殊细胞，之间的连接称为**突触**，每个神经元的计算依靠电流进行，这些电流由突触电信号引发，在神经元中央部分进行整合，并且在超过兴奋阈值时将电脉冲传递给其他神经元。神经元和突触在第 4 章中已经展示过了（见图 4-3）。为神经元建模的一个方法是线性分类器，它判断输入的加权和是否通过某个“挤压”函数（见图 4-4）。这个挤压函数的输出水平用以表示神经冲动的频率，范围从零到最大频率。

因此，单一的神经元是**一个简单的计算单元**，它计算一个标量积，接着是一个 S 型函数。顺便说一下，这个计算相当嘈杂和不规则，因为它是基于电信号的，而这个电信号又受到化学物质、压力、血液供应、血糖水平等因素的影响。神经系统的智能是按照互联的强度编码的，并且**通过改变连接来进行学习**。这种模式与“标准”串行计算机十分不同，串行计算机以周期的方式执行：从存储器中取出内容，进行数学运算，再将结果写回储存器。神经网络不区分存储器和处理过程，而是通过网络中的信号流来操作。

主要的待解之谜是：许多简单的单位连结起来的系统，竟可以产生如此令人难以置信的智能活动，比如识别物体、说话、听懂别人说话、喝一杯咖啡，以及为了你的职业生涯而努力。**涌现**（emergence）是**多个相对简单的交互形成复杂系统**的方式。类似的涌现特性在自然界也

可以观察到，想想雪花复杂的对称图案，也是从简单的水分子开始形成的。

因此，真正的大脑是研究者不可思议的灵感来源，同时也**证明了十分简单的互联计算单元能组成智能系统**。早在计算机发展的初期，这个生物领域的隐喻就已经很诱人了（“电子大脑”），但是当时仅有一些简单的类比，并没有构造智能系统的蓝图，就像弗雷德里克 杰利内克所说的“**飞机不扇动翅膀**”。然而，在 20 世纪 60 年代以及 80 年代末，生物大脑运作的原则作为一种计算工具的理念又抬头了。思维的转变导致了研究模式的改变，从基于符号规则和推理的人工智能，转变为知识被编码在系统参数（像突触互联权重）中的人工神经系统，学习的过程就是在外部刺激的影响下逐渐修改这些参数。

由于单个神经元的功能是相当简单的，它只是用一个超平面将输入空间分成两个区域，复杂性必然来自参与一个复杂的行为的更多层的神经元（就像在所有可能的情况下认出你的祖母一样）。这里的“挤压”函数为这个系统引入关键的非线性关系，没有它们的话，再多的层也只能产生线性函数。有组织的层次是大脑皮层中真实可见的，它是大脑中控制记忆力、注意力、感性认识、思维、语言和意识的那一部分（见图 9-1）。

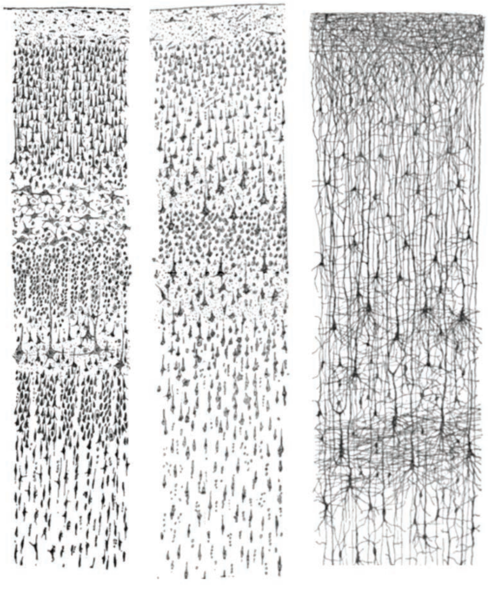


图 9-1 3 张皮层分层图，由圣地亚哥 拉蒙–卡哈尔绘制，每张都表示一个垂直方向的横截面，顶端是皮层的表面。不同的节点显示神经元的细胞体，以及一些随机神经元子集的树突和轴突

对于更复杂的“序列的”运算，例如逻辑推理，反馈回路是必不可少的，但更难以通过人工神经网络模拟。正如你所认为的那样，人工智能中“高层次”、符号化和推理的观点与“低层次”、子符号化的人工神经网络的观点是互补的。对于计算机来说很简单的任务，比如解方程或者推理，对人脑来说是困难的；对于人脑来说很简单的任务，比如认出你的祖母，仍难以

在计算机上进行模拟。现在普遍承认两种风格的智能行为，也导致关于“快思维和慢思维”的畅销书的出现 [71]。

无论什么情况下，都会有“飞机不扇动翅膀”的情况。尽管人类大脑是灵感的源泉及可行性的凭证，大多数人工神经网络却实际运行在标准计算机上。诸如“神经网络”“机器学习”“人工智能”等领域事实上逐渐融合起来，这些不同的术语各自涵盖一系列技术，这些技术用于智能系统中不同且经常互补的一些方面。

本章的重点是**前馈多层感知器神经网络**（无反馈回路）。

9.1 多层感知器

8.1 节中的 logistic 回归模型，通过将 S 型传递函数应用到无限制的线性模型输出上，使得输出可以被解释为一个概率值，它是添加“最小限度的非线性”的简单方法。可以把 logistic 回归模型看作把划分输入空间的一个刚性平面（基于线性计算和阈值的比较，一边输出 0，另一边输出 1）转变成平滑的灰色过渡区，这个平面的两边，一端是越远离平面就越黑，而另一端是越远离平面就越白，两端的中间是灰色的 ①（见图 9-2）。

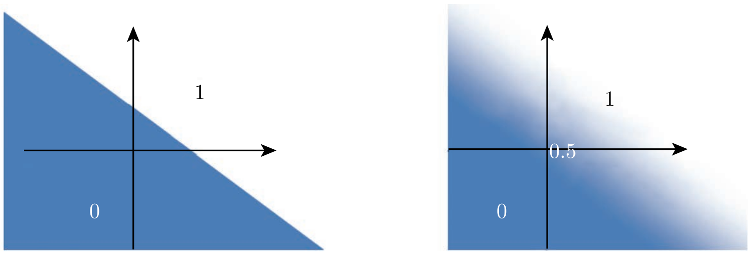


图 9-2 logistic 函数的效果：带有阈值的线性模型（左）；S 型平滑过渡（右）

如果将 y 视作地形的高度，那么许多情况下，山区有太多的丘陵、山峰和山谷，因此无法用一个平面或者单一的平滑过渡区来建模。

如果将线性变换接连组合起来，情况也不会改变：两个连在一起的线性变换仍然是线性的 ②。但是如果将第一个线性变换的输出进行非线性 S 型函数变换，然后再进行第二个线性变换，就可以得到想要的结果了：**能够逼近所有光滑函数的灵活模型**。术语**非参数模型**用于强调它们的灵活性，也用于将它们与刚性模型区分开来。在刚性模型中，只有某些参数可以根据数据进行调整。参数化模型的一个例子是振荡 $\sin(\omega x)$，其中参数 ω 必须由实验数据确

① 观察一下，可以注意到 logistic 回归与没有隐藏层却有单一输出值的 MLP 网络确实有着相同的体系结构，改变的只是进行优化的函数、MLP 优化误差平方和，以及针对 logistic 回归优化的对数似然函数（LogLikelihood）。

② 考虑一下两个线性变换 A 和 B。在 A 之后应用 B，得到的 $B(A(\boldsymbol{x}))$ 仍保持线性性。事实上，$B(A(a\boldsymbol{x}+b\boldsymbol{y})) = B(aA(\boldsymbol{x})+bA(\boldsymbol{y})) = aB(A(\boldsymbol{x}))+bB(A(\boldsymbol{y}))$。

定。第一个线性变换将提供输出的第一个“隐藏层”(隐藏是因为它处于内部，而且不能像最终输出那样直接可见)，第二个变换将从隐藏层中产生可见输出。

多层感知器神经网络是由大量的高度互联单元(神经元)构成的，它们平行工作，用于解决特定的问题。神经元以层的方式组织起来，之间有前馈信息流(无回路)。图 9-3 展示了该体系结构。

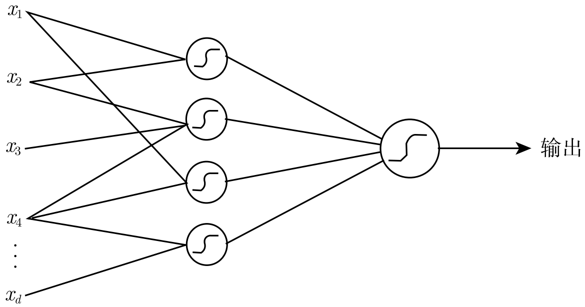


图 9-3　多层感知器：中间(隐藏)层的 S 型传递函数引入的非线性，使得创建任意连续函数变成可能。图中展示了单隐藏层

多层感知器的体系结构组织方式如下。信号从输入层依次流过不同的层，最后到达输出层。中间层称为隐藏层，因为它们在输入端或输出端是不可见的。对于每个层，每个单元首先计算权向量与另一个向量之间的标量积，这个向量是由前一层的输出给出的。得到的结果经过一个*传递函数*，产生下一层的输入。一个常用的光滑渐近传递函数(输入是大的负信号时，输出趋近 0；输入是大的正信号时，输出趋近 1)是 S 型函数，称其为 S 型函数，是因为它的图像形状像英文字母 S。之前遇见过的 logistic 转换就是一个例子(见图 8-1)：

$$f(x) = \frac{1}{1 + \mathrm{e}^{-x}}$$

其他传递函数可用于输出层。例如恒等函数，可用于无限输出值，而 S 型输出函数更适合“是/否”的分类问题或对概率建模。

关于 MLP 的一个基本问题是：**这样的结构**来表示输入–输出映射，**灵活性有多大**？换句话说，给定一个函数 $f(\boldsymbol{x})$，是否存在一个 MLP 网络和特定的权重，使得这个 MLP 的输出能够很好地逼近函数 f。虽然感知器的建模能力有限，只能用于两种模式(即输入)能被输入空间中的一个超平面分开的分类问题，但是 MLP 却是一种**通用逼近**[62]：如果有足够多的隐藏节点，拥有一个隐藏层的 MLP 能够以任何精度逼近任何光滑函数。

这是一个有趣的结果：使用与神经类似的架构，将简单的单元(线性叠加和挤压 S 型传递函数)以层的方式组织起来，并且有至少一个隐藏层，就能模拟任何光滑输入–输出的函数。

对于数学专业的同事而言，这是一个非常棒的“存在性”结果。而对于偏应用方向的同事，接下来的一个问题是：已知存在一个 MLP 逼近，怎样从已标记的实例开始快速找到它？

在阅读了上一章后，你应该已经知道至少一种可能的方法。试着想一想，然后再继续读下一节。

作为 MLP 输入–输出变换的例子，图 9-4 展示了不同的输入参数对应输出值的光滑非线性的演化。通过使用滑块，可以固定输入值的一个子集，并且两个所选输入参数的值域对应的输出用不同颜色表示。

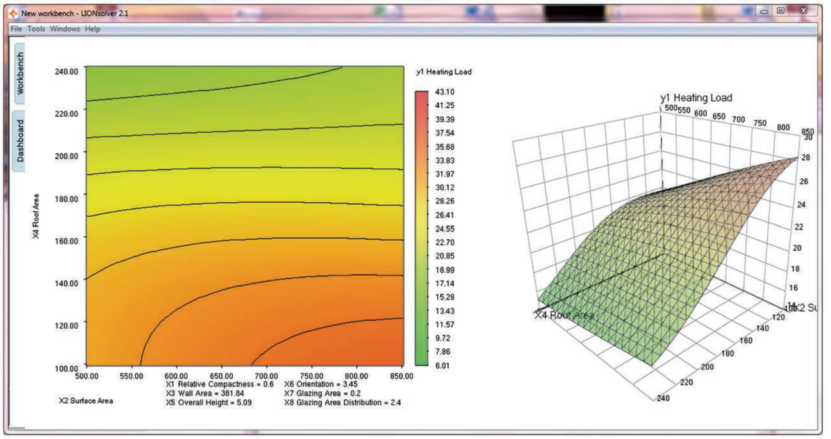


图 9-4　用 LION 软件 Sweeper 分析神经网络的输出。输出值和冬季加热房子消耗的能量，是输入参数的函数。图中展示了颜色编码的输出（左）和表面图（右）。非线性是可见的（另见彩插）

9.2　通过反向传播法学习

跟往常一样，选择一个“向导”函数进行优化，像传统的训练实例上的平方误差和，保证它是光滑的（可导），并使用**梯度下降法**。迭代地计算函数关于权重系数的梯度，并且朝着负梯度的方向移动一小步如果梯度不是 0，那么就存在一个足够小的步子，沿着负梯度的方向，能够使得函数值减小。

现在的技术问题就是使用微积分中的**链式法则**来计算偏导数，以求得两个或更多函数的复合函数的导数。如果 f 和 g 分别是一个函数，然后链式法则指明如何从 f 和 g 的导数计算出复合函数 $f \circ g$ 的导数。例如，$(f \circ g)(x)$ 的链式法则是：

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\mathrm{d}f}{\mathrm{d}g} \cdot \frac{\mathrm{d}g}{\mathrm{d}x}$$

MLP 中的基本函数是：标量积，然后是 S 型函数，接下来又是标量积，如此反复一直到输出层，计算误差。对于 MLP 网络，其梯度可以被高效地计算，它的计算需要的操作数正比于权重系数的数量，实际计算中所用的简单公式与向前传递（从输入到输出）的类似，只是现在的方向不一样，从输出误差到输入。神经网络中的一个流行的技术是通过误差的**反向传播**进行学习，它刚好包含在上面提及的练习里：梯度计算和沿着负梯度的小步移动 [115, 116, 92]。

令人惊奇的是，梯度下降法的一个直接应用，直到 20 世纪 80 年代末才被广泛使用，并且为使其流行起来的研究人员带来了如此高的声望。一个可能的原因是，梯度下降法通常被认为是一个“普普通通”的方法，只能达到局部最优点（梯度为 0），并不能保证是全局最优解。因此，用随机的小的权重系数初始化网络后，需要在不同问题上进行实验，来证明梯度下降对于训练 MLP 的现实可用性。另外，记住 ML 的目标是泛化，对于这一目标，全局最优并非必要。全局最优甚至会适得其反，并且导致过度训练。

带有简单和局部机制的逐步适应的使用与神经系统有着紧密的联系，虽然真正的神经元如何具体实现梯度下降算法仍然是一个研究课题。

注意，网络训练完之后，从输入开始计算输出需要一些简单的运算，次数正比于权重系数的个数，因此如果权重系数的个数是有限的，这一操作就可以非常快地完成。

接下来简要地定义符号。考虑“标准的”多层感知器构架，只有相邻的层之间有权重系数，差平方和能量函数定义为：

$$E(w) \;=\; \frac{1}{2}\sum_{p=1}^{P} E_p \;=\; \frac{1}{2}\sum_{p=1}^{P}(t_p \;-\; o_p(w))^2 \tag{9.1}$$

其中 t_p 和 o_p 分别是模式 p 的目标值和当前输出值。S 型传递函数是 $f(x) = 1/(1 + \mathrm{e}^{-x})$。

现在可以用在某个范围内随机分布的初始权重系数来初始化。选择一个初始化范围，像 $(-0.5, 0.5)$，并不是很容易的工作，如果权重系数太大，标量积将处于 S 型函数的饱和区域，导致梯度接近于零以及数值问题。

在下面的章节中，我们将展示两个“基于梯度的”技术：标准的批量反向传播和一个带有自适应学习速率（bold driver BP，见参考文献 [4]），以及在线随机反向传播 [92]。

9.2.1 批量和 bold driver 反向传播法

批量反向传播法（batch backpropagation）是梯度下降法的一个教科书版本。在得到了梯度中所有偏导数后，记为 $g_k = \nabla E(w_k)$，下一次迭代 $k+1$ 的权重系数被下面的式子更新为：

$$w_{k+1} = w_k \;-\; \epsilon\, g_k \tag{9.2}$$

以前的更新，具有固定的*学习率* ϵ，可以看作是粗糙版本的*最速下降*（steepest descent），每次迭代都搜索沿梯度方向上的确切最小值：

$$w_{k+1} = w_k - \epsilon_k g_k \tag{9.3}$$

$$\text{其中} \quad \epsilon_k \quad \text{最小化} \quad E(w_k - \epsilon g_k) \tag{9.4}$$

对于一个特定的学习问题，如何挑选一个合适的学习速率是个很现实的问题。学习速率不应该太小，避免学习时间过长（每次迭代权重系数的改变都很小），学习速率也不应该太大，避免震荡导致的能量函数疯长（应该记住，只有在步子很小时，沿着梯度方向的改变才能保证函数值减少）。

有一个启发式的建议，可以避免这样的选择，它在学习任务运行时改变学习速率。这一方法称为 bold driver（BD），参考文献 [4] 对它有所描述。如果连续的步骤使得能量降低，那么学习速率就指数式增加。如果遇到了一个“意外”（如果 E 增加），那么学习速率就迅速减小，直到找到一个合适的值。从一个很小的学习速率开始，它的改变用下面的式子描述：

$$\epsilon(t) = \begin{cases} \rho\,\epsilon(t-1) & E(w(t)) < E(w(t-1)) \\ \sigma^l\,\epsilon(t-1) & E(w(t)) > E(w(t-1))\ \text{使用}\ \epsilon(t-1) \end{cases} \tag{9.5}$$

其中 ρ 接近于 1（$\rho = 1.1$），是为了避免频繁的“意外”，这些情况下能量值的计算都被浪费了，σ 的选择应满足快速减小（$\sigma = 0.5$），l 是能成功减少能量的减小率 $\sigma^l\epsilon(t-1)$ 的最小整数。

这种自适应 bold driver 反向传播的表现接近于（通常也优于）适当选择一个固定的学习速率所得到的。然而，作为最速下降的一个简单形式，这些技术也受到使用梯度作为搜索方向的技术的共同限制。

9.2.2 在线或随机反向传播

由于能量函数 E 是许多项的和，每一项对应一个模式，因而梯度将是相应的局部梯度 $\nabla E_p(w_k)$ 的和，$\nabla E_p(w_k)$ 是第 p 个模式中误差的梯度 $(t_p - o_p(w))^2$。

如果某人有上百万训练实例，首先对贡献 $\nabla E_p(w_k)$ 进行求和，然后走一小步。

于是，马上会想到：在计算一个 $\nabla E_p(w_k)$ 后立即沿着负的方向走一小步会如何呢？如果这一步非常小，得到的权重与初始的差别将很小，并且接下来的梯度 $\nabla E_p(w_{k+j})$ 将非常类似于原始的那些 $\nabla E_p(w_k)$。

如果以随机的顺序选择模式，可以得到所谓的**随机梯度下降**，又称作**在线反向传播法**。

顺便说一句，因为生物神经元不是很擅长复杂和长期的计算，所以在线反向传播具有多种生物学的意义。例如，如果一个孩子正在学习识别数字，当他犯了一个错时，应该立即纠正，而不是等收集了成百上千的错误之后再纠正。

在线随机反向传播的更新由下式给出：

$$w_{k+1} = w_k - \epsilon\,\nabla E_p(w_k) \tag{9.6}$$

其中模式 p 是每次迭代时从训练集中随机选择的，ϵ 是学习速率。许多情况下，这种形式的反向传播已经成功使用，条件是用户选择了适当的学习速率。该方法的主要困难是，该迭代过程不保证收敛，并且使用梯度作为搜索方向对一些问题 ① 是非常低效的。相对于批量反向传播，也就是 E 的完整梯度被用作搜索方向，这种在线方法的竞争优势在于，局部梯度 $\nabla E_p(w_k)$ 只需要单一的向前和向后传递，因此该方法的不精确性可以通过单次迭代所需的较低计算量进行补偿，特别是训练集很大并且由冗余模式组成时。这些情况下，如果学习速率是合适的，收敛的实际 CPU 时间可以是很少的。

① 精确的说法是病态问题。

小批量 BP 是批处理和在线版本之间的第三个折中选项。这种情况下，仅仅在一个随机大小为 B 的模式子集（批）上运行向前和向后传播来积累部分梯度。每 B 个向前传播修改权重。当然，极端的情况是，B 等于 1 时相当于在线 BP，B 等于模式的总数量时相当于批量 BP。

学习速率必须精心选择：如果 ϵ 过小，即使训练时间增加，也不会产生更好的泛化结果；而如果 ϵ 增大超过某一定点，振荡会逐渐变得剧烈，并且所获得泛化的不确定性会增加。

9.2.3　训练多层感知器的高级优化

认识到优化对于机器学习的重要性之后，研究人员就开始使用优化的相关文献中提到的技术，即在搜索过程中使用**更高阶导数**的信息，*而不仅仅是梯度下降*。共轭梯度法和“割线”法是两个例子，即仅使用梯度信息，以迭代的方法更新（黑塞矩阵的）二阶导数的近似值。实际上，众所周知的是，如果黑塞矩阵的*条件数*很大，使用梯度作为当前搜索方向的收敛速度会非常慢。形象的说法是，这对应于搜索空间里由“狭窄山谷”导致的曲折的搜索路径，见图 21-9。基于二阶信息的技术在神经网络社区广泛使用，它们的效用已经得到认可，特别是权重系数数量有限（<100）并要求高精度输出值的情况。参考文献 [5] 中列出了部分书目和不同的二阶技术之间的关系。两种使用二阶导数信息（以间接和快速的方式）的技术，即共轭梯度法和具有快速线搜索的一步割线方法（OSS），在参考文献 [5] 和 [4] 中有所描述。更多细节将在本书有关连续函数优化的第 21 章加以探讨。



梗　概

创建基于“真实神经网络”的人工智能是人工神经网络研究的课题。**多层感知器神经网络**（MLP）是一种灵活的（非参数）建模架构，由 S 型单元的层组成，仅相邻层之间以前馈方式相互连接起来。识别你的祖母出现在图像中的概率的单元，可以用我们的神经硬件（毫无疑问地）建模成一个 MLP 网络。人们可以通过梯度下降法的变形从已标记的实例中进行有效的培训，这一方法通常称作“误差反向传播”。作为优化方法，梯度下降的弱点并不会影响实际的效果。

人类学习和机器学习模式之间的确有着惊人的相似之处。尤其注意一点，在训练过程中越努力，提高泛化能力方面所得到的回报就越多。一个严厉的老师（在黑板上写多样化的测试题，要求你做笔记，而不是提供预习材料）可能在训练中使你痛苦，但会增强你日后人生中的精神力量。德国哲学家黑格尔在定义哲学的作用时使用了术语Anstrengung des Begriffs（“定义概念所做的努力”）。

第 10 章　深度和卷积网络

单走一步不会在地球上走出一条路，单一的思想也不会在头脑中连成路径。要走出一条很深的路，我们要一次又一次地走。要走出一条深刻的精神之路，我们必须一遍又一遍地思考，我们希望能支配我们生活的那种想法。

—— 亨利　大卫　梭罗



现在机器学习正经历着一场软革命，很久以前诞生的想法正迎来第二次青春。深度学习和卷积网络是有前途的方向，但我们也会考虑其他替代品和重要方向，例如后续章节会提到的储备池和超限计算。

有这样一则轶事：杰弗里　E. 辛顿教授率领的研究生团队在最后一分钟决定参加一场比赛，他们使用一种深度学习系统，该系统的开发没有用到特定的领域知识，却在 2012 年赢得了最高奖项。该系统可以预测哪些分子最有可能是一种有效的药物。今天许多高级的计算机视觉和语音识别应用是基于深度网络的。

这一章介绍**深度神经网络**（deep neural network）和**卷积网络**（convolutional network）。深度网络的长期目标是完全自动从大量的数据（包括已标记和未标记的）中直接开发智能系统，而不用在训练系统之前由人类专家手动提取有用的特征。总体计划是有一个分层次的前馈网络，它是自组织的，使得前几层能够提取基本构造块（特征），在随后的层里结合它们获得越来越复杂的特征（例如，图像处理中在平移或旋转下不变的特征）。**卷积网络预布置**适用于某一领域（典型的例子是计算机视觉和语音处理）的架构，其方法是插入约束，例如感受野

(receptive field)的局部性，以及分享权重。在我们的视觉系统和图像处理系统中，基本的本地筛选操作，例如对比度增强或边缘检测，在整个图像上都有应用。如果 ML 给每个像素标识一个不同的筛选器，那么会浪费宝贵的资源，而且不要忘记这个系统是用二维结构和局部关系来处理图像的，否则会受到惩罚。

10.1 深度神经网络

神经学研究的大量证据表明，人类的大脑首先提取**有用的表示，然后逐步复杂**，如此分阶段提取更高层次的概念。为了识别你的祖母，视觉皮层首先检测简单的元素，如图像边缘(强度的突然变化)，然后逐渐识别更高级别的概念，比如眼睛、嘴巴和复杂的几何特征，它们独立于图像中的特定位置、亮度、颜色等。

虽然对于合适的逼近来说，*存在*一个隐藏层就足够了，但这并不意味着*建立*这种逼近是容易的，也不意味着只需要少量的实例和少量的 CPU 时间。除了在大脑中神经系统研究的证据，也有理论依据表明：如果考虑多个隐藏层，更容易建立某些类的输入–输出映射 [21]。

ML 研究的梦想是将实例输入有许多隐藏层的 MLP，让 MLP **自动发展出内部表示**(隐藏层单元的激活模式)。训练算法应该确定连接低层的权重(更靠近感觉输入)，使得中间层次的表示对应于“概念”，这对最终复杂的分类任务是有用的。想想看，前几层可以从数据中发展出有用的规律“金块”。

这个梦想实现起来有一定的实际障碍。当在包含许多隐藏层的 MLP 网络上应用反向传播法时，第一层权重的偏导数往往是非常小的，因此数值估计会出问题。这很容易理解①：如果改变第一层的权重，那么效果会向上传播许多层，而且它往往会与数百个其他单元的影响混淆在一起。此外，饱和的单元(输出中的 S 型函数的平坦区域)会挤压变化，使得最终输出中的效果非常不明显。前几层内部表示造成的净效应，和随机设置前几层，只留下最上面的层做一些“有用的”工作的效果差不多。从另一个角度来看，当参数的数量大于实例的数目(正如深度神经网络的情况)，过度训练将变得更加危险，因为这种情况下，网络是很容易适应训练实例的，无须提取相关规律，而这些规律却是泛化必不可少的。

在 20 世纪 90 年代，这些困难使得不少用户的注意力转向了“更简单”的模型，基于带有附加约束的线性系统，如第 11 章考虑的支持向量机。

最近，**深度神经网络**(有许多隐藏层的 MLP)的复兴和更强大的训练技巧将深度学习带到了台前，在一些充满挑战的领域分类表现出色，如语音识别、图像处理，以及药物应用中的分子活性。**不用任何***特别设计的***特征方法**(根据专业领域知识和初步实验来手动调整新特征)，深度学习就可以取得令人满意的结果，以及技术层面的显著提升 [21]。

下面是深度学习的最新应用的主要方案：

① 如果你对偏导数不熟悉，想想如果权重有少量的改变(Δw)，那么输出会有什么改变(Δf)。偏导数是比值 $\Delta f/\Delta w$ 在 Δw 趋近于零时的极限。

(1) 使用许多未标记的实例进行无监督学习，以此准备初始状态下的深度网络（**无监督预训练**）；

(2) 在以无监督的方式训练了初始网络后，仅使用已标记的实例和反向传播算法做最后的调整。

对于未标记（未分类）的实例数目比已标记的实例数目大很多，并且分类过程开销较大的情况，该方案是非常强大的。举例来说，通过抓取网页收集大量的未标记图像，现在是很简单的任务了。然而，让人来描述图像内容并标记它们，开销则相对大得多。无监督系统负责提取有用的构建块，比如边缘、斑点，以及不同类型的纹理的探测器。一般来说，构建块提取自真实图像，而不是“坏的电视机屏幕”里的随机模式。

10.1.1　自动编码器

一个构建内部表示的有效的无监督方式就是自动编码器。我们建立一个带隐藏层的网络，并要求输出简单地再现输入。这乍一听上去愚蠢而无意义，但插入一个隐藏层可以完成有趣的工作，因此要求输入的原始信息被压缩成**比原信息的变量更少的编码** $c(\boldsymbol{x})$（见图 10-1）。可以肯定，这种压缩无法让所有可能输入都按原样重建。但是，这对我们的目标而言是积极的：内部表示 $c(\boldsymbol{x})$ 将被迫发掘系统的特定输入模式中的规律性，以从原始输入中提取有用和显著的信息。

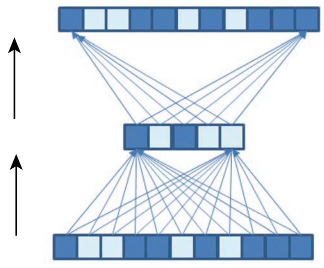


图 10-1　自动编码器

例如，如果识别脸部图像，会有一些内部单元专门检测边缘，另一些单元也许会专门检测眼睛，等等。

自动编码器可以通过反向传播及其变体进行培训。分类标签是没有必要的。如果初始输入的标记是分类，这一阶段系统只是简单地忘记标签。此外，为了训练更有健壮性，即能够更好地进行泛化，可以再添加大量未标记的实例。

自动编码器建立后，现在可以将隐藏层结构（权重和隐藏单元）移植到第二个网络进行分类（见图 10-2），添加一个附加层（以较小的随机权重初始化），并将这个“弗兰肯斯坦式怪物”网络作为训练分类器的最后阶段的起点。在这个最后阶段，只使用一组被标记的模式。

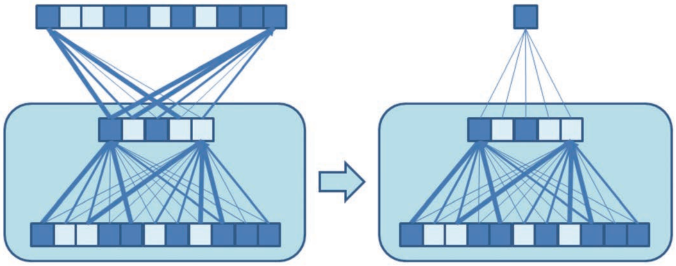


图 10-2 使用以未标记数据训练的自动编码器来初始化一个 MLP 网络

在许多重要应用中，相比于通过随机初始化所有权重获得的网络，这样最终形成的网络具有较好的泛化性能。请注意，相同的初始化正确的网络可用于不同但相关的监督训练任务。网络以相同的方式初始化，但最终调整阶段使用不同的已标记实例。将解决一个问题得到的知识应用于一个不同但相关的问题，称为**迁移学习**。例如，识别人脸的知识也可以用于识别猴子。

细心的读者可能已经注意到，到现在为止只创建了一个隐藏层。但是，我们可以很轻松地通过迭代链式地产生后续层，压缩第一个编码 $c(\boldsymbol{x})$，再自编码，得到第二个更为压缩的编码和内部表示 $c'(c(\boldsymbol{x}))$。同样，自编码的权重可以用来初始化网络的第二层，等等（见图 10-3）。

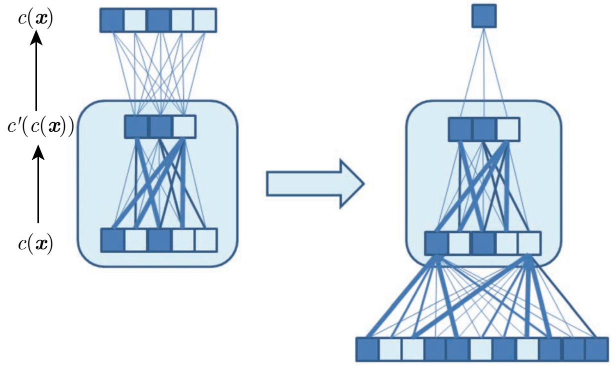


图 10-3 迭代训练自动编码器来构造更深的网络

除了用于预训练神经网络，层次**很深的自动编码器还可以用于可视化和聚类**。例如，路透社新闻故事[①]表示为 2000 个最常见单词词根文档特定的概率向量，可以自动编码，使瓶颈压缩层只含有两个单元。对应于故事的二维坐标可见于图 10-4 的二维平面。不同的聚类近似地对应于不同的标题，这是在二维空间明确可见的，因此两个（或多个）坐标可以是聚类对象很好的出发点。

① 路透社语料库卷 2 可以在 http://trec.nist.gov/data/reuters/reuters.html 找到。

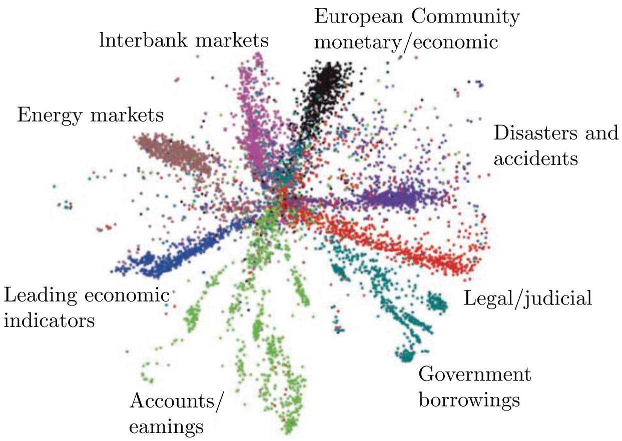


图 10-4 代码由一个 2000-500-250-125-2 自编码器根据路透社的新闻故事生成。图中用不同的颜色对应于不同主题的聚类，这是清晰可见的（详见参考文献 [57]，另见彩插）

层的最佳数目和“金字塔”结构中的单元的最佳数目仍然是一个研究课题，但是可以用务实的方法来得到合适的数目，通过使用某种形式的交叉验证来选择合适的元参数。详情见参考文献 [21]。

10.1.2 随机噪声、屏蔽和课程

结合无监督的预训练与有监督的最终调整以获得越来越深的网络，现在你该为这一想法感到兴奋，这样做能使最先进的性能需要的手工特征设计越来越少。现在讨论一些更先进的可能性，目前这些尝试已从纯粹的研究转移到现实应用。

第一种可能性是注入可控制范围内的噪声到系统 [111]（**去噪自动编码器**）。初始想法很简单：给每个模式 $\boldsymbol{x}$ 加入随机噪声（例如，如果模式是二进制的，以一个给定的小概率值翻转每个二进制位），并在自编码网络里重构原始无噪声模式 $\boldsymbol{x}$，给已污染的输入去噪。这项任务变得更加困难，但鼓励系统努力从输入模式中提取更强大也更显著的规律。这个版本弥补了与深度信念网络（DBN）之间的性能差距，并且在一些情况下还能超过它。深度信念网络是另一种预训练网络的方式 [56, 57]。从生物学的角度来说，在潮湿的大脑物质中，的确有很多的噪声。这些结果表明，噪声事实上能够对学习产生积极的影响！

还有一种方法，通过**随机屏蔽**[58]，学习的问题会更加困难，但泛化的性能会更好（通过减少过度拟合）：在随机反向传播的训练中，展示了每个训练实例后，每个隐藏单元随机在网络中忽略的概率是 0.5。这种方式避免了训练数据之间复杂的互相适应。各个单元不能依靠此处其他的隐藏单元，并且最好成为识别有用信息的探测器，这与其他单元在做什么无关。

有趣的是，屏蔽和性在进化中所起的作用有某种奇妙的相似性。一种可行的解释是，性打破了共适应基因的集合。使用一大组共适应基因实现一个功能，不如使用多个替代方式实现同样的功能那样健壮，其中每个方式只使用少量的共适应基因。这能够使得进化避免走进

死胡同，如果走进死胡同的话，适应度的改进需要协调大量共适应基因的变化。它也减少了环境中小的变化导致适应性急剧下降 —— 类似于 ML 领域里的“过拟合”现象 —— 发生的概率 [58]。

某种程度上，随机丢弃一些单元与在不同的时间使用不同的网络体系结构进行训练相关，然后在测试期间计算结果的平均值。集成不同网络是减少过度训练并提高泛化的另一种方式，会在后面的章节中加以说明。通过随机屏蔽，不同的网络被包含在一个相同的完整 MLP 网络里（通过激活完整网络的选定部分得到）。

训练 MLP 时，改进最终结果的另一种可能性是利用**课程学习**[22]。正如人们学习的时候，训练实例并不是一次性提供给神经网络，而是分步的，首先从最简单的实例开始，然后才是比较复杂的。例如，学习音乐的时候，首先是了解单个音符，然后才是更为复杂的交响乐。通过自编码来进行预训练，可以被当作课程学习的初步形式。类比于语言学习，首先学习者接触大规模的某种语言的口语材料（例如，观看某种外语电视频道）。当耳朵经过训练，适应了这门外语的发音特征之后，就为接下来更为正式的语句翻译训练做好了准备。

总之，一边睡觉一边听录音来学习语言的魔法系统也许不完全是骗局。

10.2 局部感受野和卷积网络

高等动物的大脑能快速学习处理图像并辨识其中的内容。婴儿在出生后的第一天就已经能认出自己的母亲了。如果没有已经适用于处理二维图像的**预布置的体系结构**的帮助，这样的辨识速度几乎是不可能达到的。局部性尤其起着重要作用：前面的神经元有邻近感受野，处理投影到视网膜中的图像的邻近点，并且映射到视觉皮层中的邻近点。生物大脑中存在特定的低等探测器，例如边缘探测器或运动探测器。

例如，当分析蛙类的“开–关”神经节细胞时，这种细胞对从亮到暗和从暗到亮的变化都有反应，而感受野非常受限（大约在可捕猎距离内的一只苍蝇的大小），因此难免得到这样的结论：开–关单元与刺激相匹配，并充当苍蝇探测器的职能 [3] (见图 10-5)。

当考虑人工神经网络时，基本可以确认，如果图像处理的一些知识预布置在神经网络中，那么图像识别可以大大简化。只有受虐狂才会忘记图像的二维结构，只提供位置随机分布的像素值的一维数组作为输入（如果不相信，对此页的像素做一个随机排列，然后试着读读）。

在模式识别的传统模型中，手动设计的特征从输入中提取相关信息，并消除不相干的变量。于是，像 MLP 那样可训练的分类器可以将生成的特征向量分类。一个潜在的更有趣的方案是去掉特征提取器，将原始输入给网络，再用反向传播将前几层变成相应的特征提取器。这种蛮力的方法由于输入维数非常大而面临困难（导致权重很大，并可能过度训练），同时也缺少有关输入的平移、旋转或局部扭曲的任何**内置不变性**。对于青蛙来说，一只苍蝇仍然是一只苍蝇，即使经过旋转和平移。



图 10-5　大蟾蜍，常见的蟾蜍，曾用于蟾蜍形体视觉研究。蛙类视网膜中的特征探测器是硬连线的，并且能专门检测到可以捕捉的距离内的苍蝇

原则上讲，一个足够大的全连接网络可以学习产生输出，这些输出相对于这样的变化是不变的。然而，学习这样的任务可能会导致多个单元中，权重相似的图案出现在输入的各个位置。在**卷积神经网络**（convolutional neural network，CNN）中 [76]，一些平移不变性由自动强制复制跨空间的权重配置获得。图像平面中具有本地连通性的核在图像的不同位置重复出现（**权重被共享**）。由于局部相关性的原因，识别空间或时间对象之前，就能够提取它们的局部特征并使其相结合，这也是一个比较广为人知的优势。卷积网络将隐藏单位的感受野限制在局部，以此迫使其提取局部特征。

卷积就是对于函数的不同空间位置进行相同的局部过滤的数学算子。信号处理就是一个典型的利用局部核来提取局部特征的例子，卷积是一种非常重要的算子，图 10-6 展示了两个具有代表性的例子。左图是用高斯核卷积过滤噪声信号，其输出是原始信号的平滑形式。这些步骤分别是模糊化（计算机视觉）、低通滤波或去噪（信号处理）、平滑化。在数学上，给定一个信号函数 $s(t)$ 和一个过滤器 $f(t)$，卷积算子可以写成：

$$s * f(t) = \int_{-\infty}^{+\infty} s(x)f(t-x)\mathrm{d}x \tag{10.1}$$

换句话说，过滤核 f 使用加权积分“扫过”整个信号函数，如果高斯核拥有单位面积，卷积的结果就是原信号的加权平均。图 10-6 的右图是一个更有趣的例子，它使用两个不同振幅的高斯核的差分作为滤波器，通过将原始信号的两种使用不同光滑窗口的模糊化结果相减得到。最终的核被称作**高斯差分**（DoG），能高亮那些隐藏的、突变的信号中的点。

对于神经网络，卷积公式 (10.1) 可以很容易地拓展成二维离散形式。为了实现这个目标，令 x_{ij} 为一个 $m \times n$ 图像中的像素，过滤核表示成一个 $(2r+1) \times (2r+1)$ 的元素为 w_{ij} 的矩阵，通常它的半径r 是非常小的。卷积产生一个新的 $m \times n$ 图像，它的每个像素 y_{ij} 是

$$y_{ij} = \sum_{h=0}^{2r} \sum_{k=0}^{2r} w_{hk} x_{i+r+1-h, j+r+1-k} \tag{10.2}$$

上式假设索引从 0 开始。为了得到和原图像大小相等的结果，我们必须假设原图像在任何方向都有一个大小为 r 的边界；否则，最终生成的图像在任何方向都比原图像要少 r 个像素。

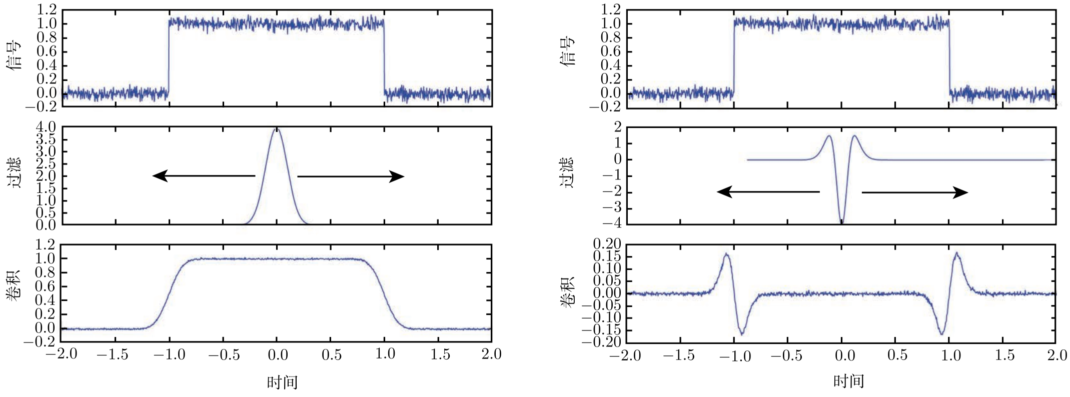


图 10-6 卷积的两个例子：（左）高斯平滑；（右）高斯差分边界增强

观察式 (10.1) 中的“$t-x$”：为了保留许多有用的数学性质，卷积运算符要求两个函数从反方向扫过，如式 (10.2) 中所再现的。大多数软件包可以用这种方式配置，也可以让核和输入从相同的方向扫过。在后一种情况下，网络以互相关模式进行操作，其结果是一个适当的内积。两种模式之间的唯一实际差别是权重被存储的顺序，在两个表示之间的变换仅需要核的 180° 旋转。换言之，卷积层求线性过滤器和相关感受野的内积。

卷积网络结合 3 种成分以确保一定程度上的移位和变形后的不变性：**局部感受野**、**共享权重**（或权重复制），以及有时会需要的空间或时间子抽样（**池化**）。

如图 10-7 所示，通过在整个图像中应用相同的局部感受野，神经元可以提取基本视觉特征，如有向边、端点、边角或语音谱图中的类似模式。然后这些特征由较高层的神经元结合起来。

带有共享权重的神经元的输出，在图像中不同的点上重复，被称为**特征映射**。特征映射是通过在输入的每个局部的卷积后接一个非线性激活函数而得到的。在图 10-7 的上半部分，输入的图像被两个过滤器“扫过”，生成两个特征映射，这两个特征映射的神经元或多或少由局部感受野中相应特征的存在而激活。在该示例中，一个过滤器专门用来辨识斜线，另一个则学会了用高斯差分来加强边界。

通常情况下，每个**卷积层**后都附加**池化层**（见图 10-7 底部），池化层执行局部平均和子抽样并降低特征映射的分辨率，因此降低了输出对于变化和扭曲的敏感度。其基本形式为池化层将每个特征层分为非重叠的矩形，并应用一个简单的“总结”操作到每个矩形像素上。常见操作如下：

- 矩形中所有像素的最大值（max-pooling）；
- 矩形中像素值的平均值（average-pooling）；
- 所有像素值的平方和的平方根（即该矩形的弗罗贝尼乌斯范数）。

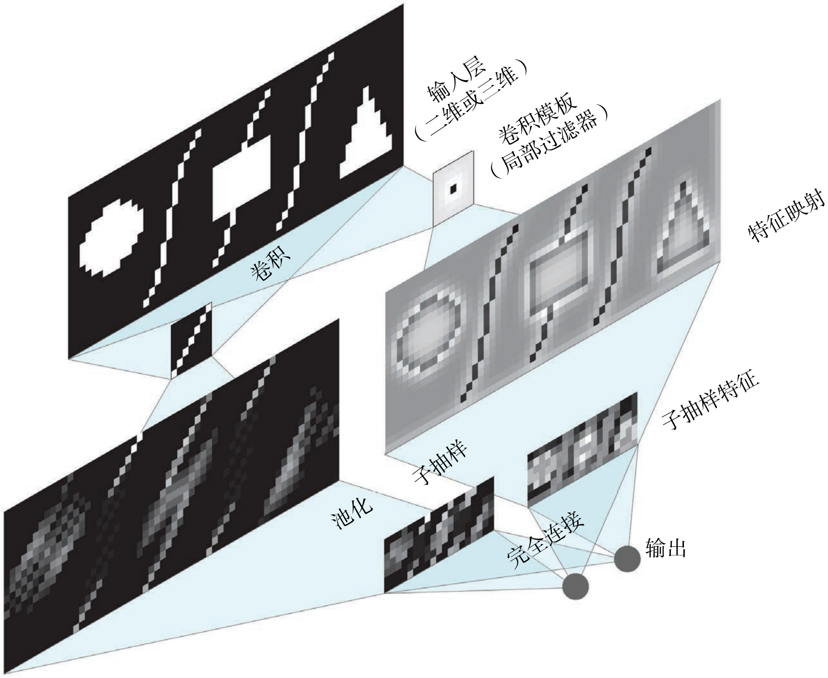


图 10-7 基本卷积网络：输入（图像像素）被针对一个输入权重的小型集合的卷积运算扫过，该卷积运算充当局部特征提取器。将所得的特征映射通过一个池化操作进行子抽样，而较小的神经元集合通过一个传统的完全连接输出层传递

更深层次的体系结构可以实现卷积层和池化层的级联，可通过其他方式提高输出的健壮性来达到，例如 10.1.2 节中讨论的随机屏蔽技术。只要神经元数目足够小，完全连接的前馈层就完成了网络。

卷积神经网络仍然是一个研究热点，并且是处理复杂图像和语音任务的先进技术，其范围过于广泛，因此本书不可能面面俱到。分层和结构化的体系结构的一个示例见于图 10-8，基于参考文献 [53] 最近提出的随机创建神经元前几层的权重。参考文献 [78] 的作者提出了一种“分形”的嵌套网络的体系结构，建立带有更复杂的结构的微神经网络，抽象出感受野内的数据（超越传统 CNN 中过滤系数和图像像素之间的线性标量积），然后在整个图像上复制该微神经网络（micro neural network，MLP）。

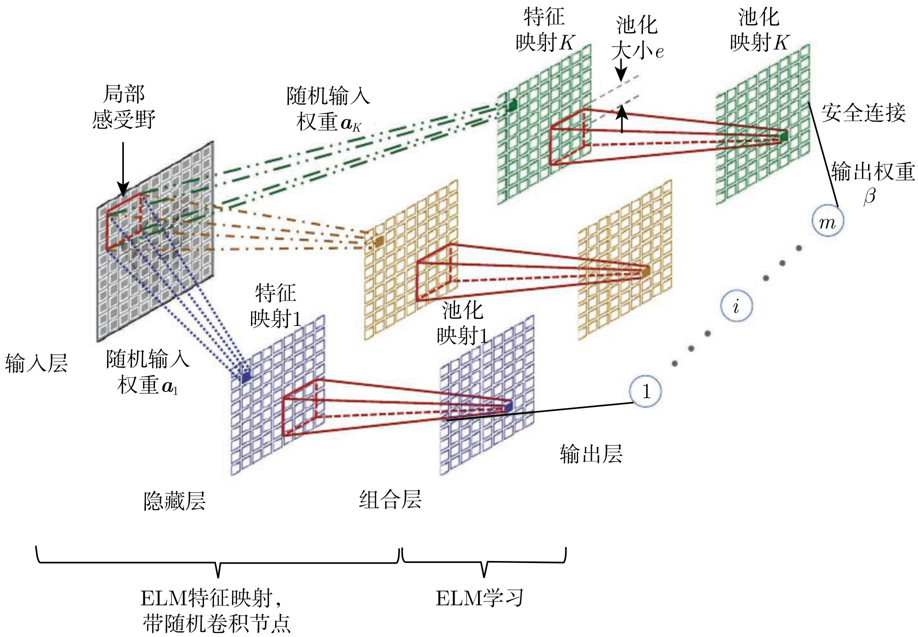


图 10-8　一个带有局部感受野的结构化的体系结构（卷积）和池化层（改编自参考文献 [53]）



梗　　概

通过使用适当的学习模式，多层**深度神经网络**会变得更有效率（且优于支持向量机），包括无监督的筹备阶段以及之后的最终调整阶段，在最终阶段需要利用稀缺的已标记实例。

在改善泛化能力的方法中，**在训练中使用可控制的噪声的方法**是有效的（噪声自动编码器和随机屏蔽）。如果你觉得大脑里有噪声和混乱，请放松，它们也许是有好处的。

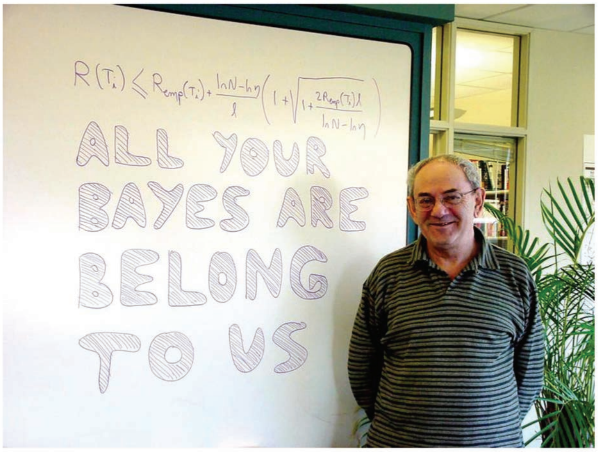
卷积神经网络是从生物学得到灵感，并具有工程上的竞争力的想法的一个很好的例子，而且它提倡构建嵌入专业领域知识的**预布置的体系结构** 。

神经网络就像一个冰川湖。你潜入水中，但你不知道下面会有多深。

第 11 章　统计学习理论和支持向量机

看上去是困难，事实上却是机会。

—— 詹巴蒂斯塔　维科



本书中各章节的次序在某种程度上是按照机器学习的历史发展排列的。① 在 1980 年之前，大部分的学习方法集中于基于规则符号的专家系统，或精粒度亚符号的线性判别技术，这些技术都有着明确的理论属性。到了 20 世纪 80 年代，决策树和神经网络为非线性模型提供了有效的学习方法，但却缺少坚实的理论基础，而基于梯度下降的最优化技术也略显朴素。

到了 20 世纪 90 年代，得益于 Vapnik 和 Chervonenkis 的开创性工作，研究者为非线性函数建立了许多基于统计学习理论的有效学习算法。**统计学习理论**（SLT）解答了**从数据中学习**的根本问题：什么情况下一个模型可以从样本中学习？一个模型在一组样本上测得的性能是如何约束其泛化的性能的？

这些理论结果是持久不变的，尽管这些定理的有效性在大部分的现实问题中几乎不可能得到验证。另一方面，这些研究人员计划复兴**线性判别**方法，他们为了加强模型的泛化能力，往线性判别方法中加入额外的优化目标，并把这种方法称作**支持向量机**（SVM）。

SVM 听起来很专业，但其基本原理很容易掌握。考虑图 11-1（左图）中的两类点（分别是灰色和白色），以及两条直线 A 和 B，它们都可以线性划分这些点，并分别是划分带标签训

① Vapnik 教授的照片，来自 Yann LeCun 的网站“Vladimir Vapnik 与视频游戏亚文化的相遇”，http://yann.lecun.com/ex/fun/index.html#allyourbayes。

练数据这一常见机器学习方法的两种不同结果。当我们泛化划分结果时，就能发现这两种结果的不同之处。使用这个已训练好的系统时，新的样本来自与训练样本相同的概率分布，即两类点在图中的分布与训练样本类似，但是对于直线 B，样本点落入分类器错误一侧的概率会远大于直线 A。直线 B 离一些训练样本点很近，因此几乎不能分离这些点。而直线 A 离两类样本点的概率距离都是最远的，因此在它的附近有概率上最“安全的区域”，又称作间隔（margin）。SVM 就是具有**最大可能安全间隔的线性分类器**，其中的支持向量就是那些处于安全间隔两侧边缘的点（见图 11-1 右图）。其实，我们遇到过的最小二乘线性模型（见 4.3 节）和 SVM 很相似。最小二乘法最小化均方误差，而 SVM 最小化最大距离，不过二者的目标是一致的，都是为了得到类间健壮且安全的边界。

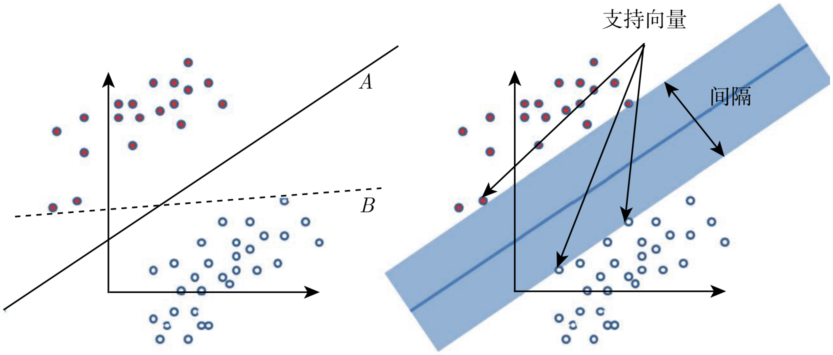


图 11-1　解释支持向量机的基础。线 A 的间隔比线 B 的要大。大的间隔会增加新实例落在分类器正确一侧的概率。支持向量是触及最大可能间隔的那些点

为了得到最大间隔线性分类器，通常使用标准**二次规划**，它可以在一定规模下解决此类优化问题。二次规划问题就是目标函数为二次函数、约束条件为线性的最优化问题。在多层感知器中存在的局部极小值问题 —— 由于局部极小值离全局最小值很远 —— 在二次规划中不会出现，因此可以放心使用 SVM。但众所周知，没有不带刺的玫瑰，如果训练样本不是线性可分的，那么 SVM 就会遇到很多问题。这种情况下，需要先对原始样本点做**非线性的变换**ϕ，从而将其变成（近似）线性可分的。可以将 ϕ 看作一个合适的特征生成函数，它使得变换之后的两类样本点 $\phi(\boldsymbol{x})$是线性可分的。对于特定的问题，需人工生成特定的非线性变换，目前还没有通用的变换。

难道为了找到合适的 ϕ，还要重新做特征提取和特征工程？某种意义上是这样的，在使用 ϕ 变换输入样本后，**SVM 的特征就是要识别的样本和训练样本**[①]**之间所有的相似性值**。SVM 关键的一步就是，通过一些交叉验证的方式，人工确定最利于学习和泛化的相似度度量函数，其中就涉及**核函数**的选择。

① 实际上只有支持向量才提供非零的贡献。

SVM 可以看作解决了两个问题：一方面，它找到了一个衡量输入向量之间**相关性**的合适方式，即核函数$K(\boldsymbol{x},\boldsymbol{y})$；另一方面，它构建了一个**线性结构**，该线性结构结合了训练样本的输出和新的测试样本，训练样本的输出用相似度来衡量。正如预期的那样，越相似的输入样本对输出的贡献越大，就像第 2 章中更原始的最近邻分类器一样，可以用类似下面的式子来描述：

$$\sum_{i=1}^{\ell} y_i \lambda_i^* K(\boldsymbol{x},\boldsymbol{x}_i)$$

（ℓ 是训练样本的数量，y_i 是训练样本 $\boldsymbol{x}_i$ 的输出，$\boldsymbol{x}$ 是待分类的新测试样本。）这个式子在下面的理论描述中会再次出现。核在计算被函数 $\phi(\boldsymbol{x})$ 映射后数据点的点积（纯量积）时，实际上不用计算这个映射函数，这种方法被称作“**核方法**”（见图 11-2）：

$$K(\boldsymbol{x},\boldsymbol{x}_i) = \boldsymbol{\varphi}(\boldsymbol{x}) \cdot \boldsymbol{\varphi}(\boldsymbol{x}_i)$$

一个包含一系列点对内核值的对称半正定格拉姆矩阵融合了数据和核的信息。[①] 为获得的数据而估计一个合适的具有最大泛化结果的核矩阵，这是一个正在开展的研究课题。

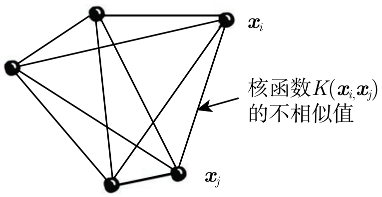


图 11-2　支持向量机学习的初始信息是每对输入点之间的相似性值 $K(\boldsymbol{x}_i,\boldsymbol{x}_j)$，其中 K 被称为核函数。这些值，在某些条件下，可以理解为初始输入通过一个非线性函数 $\phi(\boldsymbol{x})$ 的映射后得到的标量积，但并不需要计算实际的映射，仅需要计算核的值（“核方法”）

现在，SVM 的整体框架已经明确，下面就来深入数学的细节，其中有些细节非常复杂难解。幸运的是，使用 SVM 的时候，并不需要知道这些定理的证明，虽然了解主要的数学结果会帮助你更好地选择参数和核等。

11.1　经验风险最小化

之前提到过，最小化一系列样本的误差并不是一个合理的统计学习算法的唯一目标，也要考虑模型的结构。**统计学习理论**为基于观测的推导未知函数依赖关系提供了数学工具。

统计学中的**范式转换**始于 20 世纪 60 年代：在此之前，基于费希尔在 19 世纪二三十年代的研究，研究者为了从观测样本中推导出函数依赖关系，必须了解所需依赖关系的详细形式，

① 任何相似矩阵都可以被用作核，只需满足 Mercer 定理的条件。

并且从实验数据中只能得到有限数量参数的值。而新的范式不需要详细了解依赖关系，并证明了一些未知依赖关系的函数集合的某些通用属性足以估计数据的依赖关系。**非参数技术**就是这些灵活模型的一种，研究者即使不了解输入–输出函数的详细形式也能使用该方法，例如之前的多层感知器（MLP）模型。

简单总结一下统计学习理论主要方法的要点，对于促进使用支持向量机（SVM）作为一个学习机制有巨大作用。令 $P(\boldsymbol{x}, y)$ 为抽样的未知概率分布，任务是学习映射 $\boldsymbol{x}_i \to y_i$，即得到函数 $f(\boldsymbol{x}, \boldsymbol{w})$ 的参数值。函数 $f(\boldsymbol{x}, \boldsymbol{w})$ 称作*假设*，集合 $\{f(\boldsymbol{x}, \boldsymbol{w}) : \boldsymbol{w} \in \mathcal{W}\}$ 称作*假设空间*，记作 $\mathcal{H}$，令 $\mathcal{W}$ 为抽象参数的集合。一个基于标记样本选择的参数 $\boldsymbol{w} \in \mathcal{W}$ 就得到了一个“训练机”。

一个用于分类的训练机的*期望测试误差*或*期望风险*是：

$$R(\boldsymbol{w}) = \int \|y - f(\boldsymbol{x}, \boldsymbol{w})\| \mathrm{d}P(\boldsymbol{x}, y) \tag{11.1}$$

而经验风险$R_{\text{emp}}(\boldsymbol{w})$ 则是训练集上的平均误差率：

$$R_{\text{emp}}(\boldsymbol{w}) = \frac{1}{\ell} \sum_{i=1}^{\ell} \|y_i - f(\boldsymbol{x}_i, \boldsymbol{w})\| \tag{11.2}$$

一个基于**经验风险最小化**（ERM）原则的经典学习方法是：可以通过最小化经验风险 [式 (11.2)] 来逼近函数 $f(\boldsymbol{x}, \hat{\boldsymbol{w}})$，随后最小化期望风险 [式 (11.1)]，从而逼近函数 $f(\boldsymbol{x}, \boldsymbol{w}^*)$。

经验风险最小化的基本依据是：如果 R_{emp} 依概率收敛于 R（由大数定律保证），那么 R_{emp} 的最小值可能收敛到 R 的最小值。如果这个依据不成立，那么经验风险最小化原则就被称作*不一致*。

Vapnik 和 Chervonenkis 指出，上述一致性成立，当且仅当 R_{emp} 依概率收敛到 R 是*一致*的，即随着训练集的增加，$R_{\text{emp}}(\boldsymbol{w})$ 逼近 $R(\boldsymbol{w})$ 的概率对于整个参数集合 $\mathcal{W}$ 一致地趋近 1。经验风险最小化的充要条件是假设空间 $\mathcal{H}$ 的 **Vapnik-Chervonenkis 维**（VC-dimension，VC 维）是有限的。

一个假设空间的 VC 维，简单来说，就是能被函数集合 $f(\boldsymbol{x}, \boldsymbol{w})$ 分割成所有可能的两种类别的最大样本数。VC 维 h 描述了**假设空间的复杂度和表达能力**，通常与模型 $f(\boldsymbol{x}, \boldsymbol{w})$ 的自由参数的数量成正比。

Vapnik 和 Chervonenkis 规定了经验风险和期望风险之间偏离的界限，可以依概率 $1 - p$ 写成下式：

$$R(\boldsymbol{w}) \leqslant R_{\text{emp}}(\boldsymbol{w}) + \sqrt{\frac{h\left(\ln \frac{2\ell}{h} + 1\right) - \ln \frac{p}{4}}{\ell}} \quad \forall \boldsymbol{w} \in \mathcal{W}$$

通过分析这个界限并忽视对数因子，为了得到较小的期望风险，我们要使经验风险和假设空间的 VC 维与训练样本数的比例 h/ℓ 变得很小。换句话说，要想在训练之后得到有效的

泛化能力，就要使假设空间足够大，使得训练机经验风险较小，即能够正确训练样本，但假设空间又不能太大，否则会导致训练机仅仅简单地记忆训练样本，而没有提取出问题的结构。所以，为了得到更好的模型适应性，也需要更多的样本来实现类似的泛化水平。

尤其当样本数量有限制时，为了得到好的泛化效果，选择适当的 VC 维至关重要。

为了选择合适的 h 值的，Vapnik 在上述界限的基础上提出了**结构风险最小化**（structural risk minimization, SRM）的方法。对于 SRM 的原理，学习模型从一个嵌套的假设空间开始：

$$\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots \subset \mathcal{H}_n \subset \cdots \tag{11.3}$$

并且集合 $\mathcal{H}_n$ 的 VC 维 $h(n)$ 满足 $h(n) \leqslant h(n+1)$ 这样的性质。当下标数值 n 增加时，最小经验风险降低，但是关于置信区间的值会增大。SRM 原则就是选择对真实风险具有最小界限的假设子集 $\mathcal{H}_n$。暂且忽略对数因子，必须解决下述问题：

$$\min_{\mathcal{H}_n}\left(R_{\text{emp}}(\boldsymbol{w}) + \sqrt{\frac{h(n)}{\ell}}\right) \tag{11.4}$$

下述的 SVM 算法就是基于 SRM 原则，通过同时最小化 VC 维的界限和训练错误的样本数来达到。

SVM 的数学推导在线性分类问题中第一次被总结，之后也为建立其他模型提供一些直觉上的基础。

11.1.1　线性可分问题

假设这些已标记的实例是线性可分的，这意味着存在一对 $(\boldsymbol{w}, b)$ 使得：

$$\begin{aligned}
&\boldsymbol{w}\cdot\boldsymbol{x} + b \geqslant 1 \quad && \forall \boldsymbol{x} \in 类_1 \\
&\boldsymbol{w}\cdot\boldsymbol{x} + b \leqslant -1 && \forall \boldsymbol{x} \in 类_2
\end{aligned}$$

假设空间包含函数：

$$f_{\boldsymbol{w},b} = \text{sign}(\boldsymbol{w}\cdot\boldsymbol{x} + b)$$

由于参数 $(\boldsymbol{w}, b)$ 乘以一个常数不改变判定表面，下列约束可用来确定单一的一对：

$$\min_{i=1,\ldots,\ell} |\boldsymbol{w}\cdot\boldsymbol{x}_i + b| = 1$$

假设空间的结构可以通过限制向量 $\boldsymbol{w}$ 的范数引入。Vapnik 已经证明，若所有实例位于 n 维中半径为 R 的球面里且 $\|\boldsymbol{w}\| \leqslant A$，则函数集 $f_{\boldsymbol{w},b} = \text{sign}(\boldsymbol{w}\cdot\boldsymbol{x} + b)$ 的 VC 维度 h 满足

$$h \leqslant \min\{\lceil R^2A^2 \rceil, n\} + 1$$

限制 $\boldsymbol{w}$ 的范数提供假设空间的约束的几何解释如下（见图 11-3）：如果 $\|\boldsymbol{w}\| \leqslant A$，那么从超平面 $(\boldsymbol{w}, b)$ 到最近的数据点的距离大于 $1/A$，因为只考虑与在每个数据点周围半径为 $1/A$

的球不相交的超平面。在线性可分的情况下，最小化 $\|\boldsymbol{w}\|$ 来确定最大边界（两个训练类的凸包之间的沿着垂直于超平面测量的距离）的分离超平面。

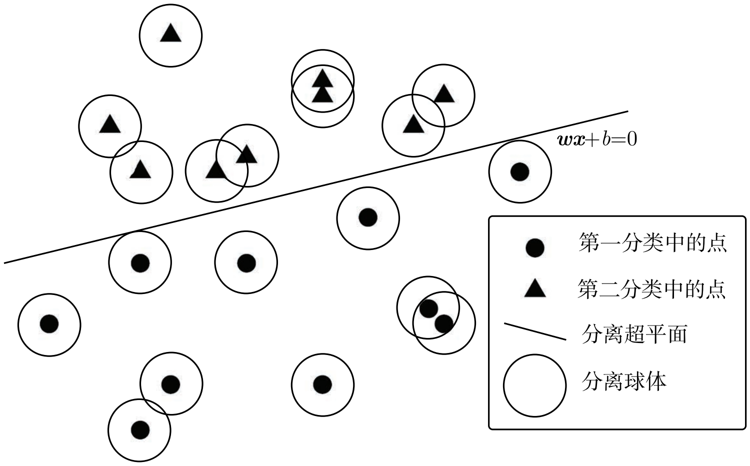


图 11-3 假设空间约束。该分离超平面必须最大化边界。直观来说，没有太靠近边界的点，使得输入数据中的一些噪声和将来由相同概率分布产生的数据不会破坏分类

这一问题可以形式化为：

$$\begin{aligned}&\text{最小化}_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2\\&\text{使服从} \quad y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i+b)\geqslant 1 \quad i=1,\cdots,\ell\end{aligned}$$

这个问题可以通过使用标准**二次规划**（QP）优化工具来解决。

引入一个向量 $\boldsymbol{\Lambda}=(\lambda_1,\cdots,\lambda_\ell)$ 作为对应于约束的非负拉格朗日乘数，那么对偶二次规划如下：

$$\begin{aligned}&\text{最大化}_{\boldsymbol{\Lambda}} \quad \boldsymbol{\Lambda}\cdot\mathbf{1}-\frac{1}{2}\boldsymbol{\Lambda}\cdot\boldsymbol{D}\cdot\boldsymbol{\Lambda}\\&\text{使服从} \quad \begin{cases}\boldsymbol{\Lambda}\cdot\boldsymbol{y}=0\\\boldsymbol{\Lambda}\geqslant 0\end{cases}\end{aligned} \tag{11.5}$$

其中 $\boldsymbol{y}$ 是实例分类向量，$\boldsymbol{D}$ 是对称 $\ell\times\ell$ 矩阵，其元素 $D_{ij}=y_iy_j\boldsymbol{x}_i\cdot\boldsymbol{x}_j$。

对应于 $\lambda_i>0$ 的向量 $\boldsymbol{x}_i$ 为支持向量。换句话说，支持向量是式 (11.5) 中的约束为活跃的那些向量。如果 $\boldsymbol{w}^*$ 是 $\boldsymbol{w}$ 的最优值，那么对于任何支持向量 $\boldsymbol{x}_i$，b 的最优解的值可以计算 $b^*=y_i-\boldsymbol{w}^*\cdot\boldsymbol{x}_i$，并且分类函数可以写成

$$f(\boldsymbol{x})=\operatorname{sign}\left(\sum_{i=1}^{\ell}y_i\lambda_i^*\boldsymbol{x}\cdot\boldsymbol{x}_i+b^*\right)$$

需要注意的是求和指数也可以限制在支持向量上，因为所有其他的向量的 λ_i^* 系数为零。最终分类是由加权的子分类 y_i 的线性组合决定的，这些权重由输入模式与实例模式的标量积（当前模式和实例 $\boldsymbol{x}_i$ 之间的“相似性”的度量）和参数 λ_i^* 确定。

11.1.2　不可分问题

如果假设集不变，但实例不是线性可分的，那么可以给引入正比于约束违反 ξ_i（收集在向量 $\boldsymbol{\Xi}$ 里），然后对以下问题求解：

$$
\begin{aligned}
&\text{最大化}_{\boldsymbol{w},b,\boldsymbol{\Xi}} && \frac{1}{2}\|\boldsymbol{w}\|^2 + C\left(\sum_{i=1}^{\ell}\xi_i\right)^k \\
&\text{使服从} && \begin{cases} y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i + b) \geqslant 1-\xi_i & i=1,\cdots,\ell \\ \xi_i \geqslant 0 & i=1,\cdots,\ell \\ \|\boldsymbol{w}\|^2 \leqslant c_r \end{cases}
\end{aligned}
\tag{11.6}
$$

其中参数 C 和 K 确定违反约束造成的成本，而 c_r 限制系数向量的范数。事实上，最小化的第一项与 VC 维度有关，而第二项与经验风险相关。（参见前文的 SRM 原理。）在本例中，k 设为 1。

11.1.3　非线性假设

上述技术可以扩展到非线性分类器，这需要将输入数据 $\boldsymbol{x}$ 映射成高维特征向量 $\boldsymbol{\varphi}(\boldsymbol{x})$ 并在转化的空间再使用线性分类，转化后的空间称为特征空间。现在 SVM 分类器变为：

$$
f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^{\ell} y_i\lambda_i^*\boldsymbol{\varphi}(\boldsymbol{x})\cdot\boldsymbol{\varphi}(\boldsymbol{x}_i) + b^*\right)
$$

引入核函数 $K(\boldsymbol{x},\boldsymbol{y}) \equiv \boldsymbol{\varphi}(\boldsymbol{x})\cdot\boldsymbol{\varphi}(\boldsymbol{y})$，则 SVM 分类器变为：

$$
f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^{\ell} y_i\lambda_i^* K(\boldsymbol{x},\boldsymbol{x}_i) + b^*\right)
$$

相对应的二次优化问题变为：

$$
\begin{aligned}
&\text{最大化}_{\boldsymbol{\Lambda}} && \boldsymbol{\Lambda}\cdot\boldsymbol{1} - \frac{1}{2}\boldsymbol{\Lambda}\cdot\boldsymbol{D}\cdot\boldsymbol{\Lambda} \\
&\text{使服从} && \begin{cases} \boldsymbol{\Lambda}\cdot\boldsymbol{y} = 0 \\ 0 \leqslant \boldsymbol{\Lambda} \leqslant C\boldsymbol{1} \end{cases}
\end{aligned}
\tag{11.7}
$$

其中 $\boldsymbol{D}$ 是一个对称的 $\ell\times\ell$ 矩阵，其元素 $D_{ij} = y_i y_j K(\boldsymbol{x}_i,\boldsymbol{x}_j)$。

SVM 方法的一个扩展是让两个类的失误有不同的权重，例如当两个类的样本容量不一样的时候，或者当发生某个类的失误比发生另一个类的失误要严重的时候。这可以通过给两个

类的失误设置不同的惩罚 (C^+ 和 C^-) 来完成。现在需要最小化的函数变为：

$$\frac{1}{2}\|\boldsymbol{w}\|^2 + C^+\left(\sum_{i:y_i=+1}^{\ell}\xi_i\right)^k + C^-\left(\sum_{i:y_i=-1}^{\ell}\xi_i\right)^k$$

如果小心选择特征函数 $\boldsymbol{\varphi}(\boldsymbol{x})$，人们无须实际计算所有特征就可以**计算标量积**，因此大大降低了计算复杂度。

避免这种显式映射的方法也被称为**核方法**。这种方法使用只需向量在原输入空间的点积的学习算法，并通过**核函数**的手段，选择使得这些高维点积可在原空间内进行计算的映射。

例如，在一维空间中的一个合理选择可以是带有适当系数 a_n 的变量 x 的单项式：

$$\boldsymbol{\varphi}(x) = (a_0 1, a_1 x, a_2 x^2, \cdots, a_d x^d)$$

这样 $\boldsymbol{\varphi}(x)\cdot\boldsymbol{\varphi}(y) = (1+xy)^d$。在更高的维度里，可以看出，如果特征是阶 $\leqslant d$ 的单项式，那么我们总是可以确定系数 a_n 使得：

$$K(\boldsymbol{x},\boldsymbol{y}) = (1+\boldsymbol{x}\cdot\boldsymbol{y})^d$$

内核函数 $K(\cdot,\cdot)$ 是规范内积在特征空间中的卷积。SVM 通常使用的内核有以下几种。

(1) 点积：$K(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}\cdot\boldsymbol{y}$；这种情况下没有映射，并且仅计算最佳的分离超平面。

(2) 多项式函数：$K(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}\cdot\boldsymbol{y}+1)^d$，其中阶 d 是给定的。

(3) 径向基函数（RBF），像高斯函数：带有参数 γ 的 $K(\boldsymbol{x},\boldsymbol{y}) = \mathrm{e}^{-\gamma\|\boldsymbol{x}-\boldsymbol{y}\|^2}$。

(4) S 型（或神经）内核：带有参数 a 和 b 的 $K(\boldsymbol{x},\boldsymbol{y}) = \tanh(a\boldsymbol{x}\cdot\boldsymbol{y}+b)$。

(5) ANOVA 核：带有参数 γ 和 d 的 $K(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{n}\mathrm{e}^{-\gamma(x_i-y_i)}\right)^d$。

当 ℓ 数值增加时，该二次优化问题需要一个 $\ell\times\ell$ 矩阵，因此随着训练集大小的增长，这种方法就会迅速变得不切实际。参考文献 [83] 介绍了一种分解方法，这个优化问题被分成一个活动集合和一个非活动集合。参考文献 [69] 的工作介绍了有效的方法来选择工作集以及减少问题的复杂度，它是利用这一事实：相对于训练数据点总数，支持向量数是很小的。

11.1.4 用于回归的支持向量

支持向量方法也可以用于回归，也就是说，从一组训练数据 $\{(\boldsymbol{x}_i, y_i)\}$ 来估计函数 $f(\boldsymbol{x})$。就像分类那样，先从线性函数的情况开始，然后考虑预处理输入数据 $\boldsymbol{x}_i$，将其映射到合适的特征空间，使得到的模型是非线性的。

为了使术语统一，线性情况可以概括为函数 $f(\boldsymbol{x}) = \boldsymbol{w}\cdot\boldsymbol{x}+b$。要解决的凸优化问题变为：

$$\begin{aligned}&\text{最小化}_{\boldsymbol{w}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2\\ &\text{使服从} \quad \begin{cases} y_i - (\boldsymbol{w}\cdot\boldsymbol{x}_i + b) \leqslant \varepsilon \\ (\boldsymbol{w}\cdot\boldsymbol{x}_i + b) - y_i \leqslant \varepsilon \end{cases}\end{aligned}$$

假定存在一个函数，以精度 ε 逼近所有数据对。

如果问题是不可解的，可以引入具有松弛变量 ξ_i, ξ_i^*（保存在向量 $\boldsymbol{\Xi}$ 里）的软边界，来应对不可行的约束，得到下列优化问题：

$$
\begin{aligned}
&\text{最小化}_{\boldsymbol{w},b,\boldsymbol{\Xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\left(\sum_{i=1}^{\ell}\xi_i^* + \sum_{i=1}^{\ell}\xi_i\right) \\
&\text{使服从} \quad \begin{cases} y_i - \boldsymbol{w}\cdot\boldsymbol{x}_i - b \leqslant \varepsilon - \xi_i^* & i = 1,\cdots,\ell \\ \boldsymbol{w}\cdot\boldsymbol{x}_i + b - y_i \leqslant \varepsilon - \xi_i & i = 1,\cdots,\ell \\ \xi_i^* \geqslant 0 & i = 1,\cdots,\ell \\ \xi_i \geqslant 0 & i = 1,\cdots,\ell \\ \|\boldsymbol{w}\|^2 \leqslant c_r \end{cases}
\end{aligned}
\tag{11.8}
$$

类似于分类的情况，C 决定函数的平直度和偏差大于 ε 的容许度之间的权衡。关于支持向量用于回归的详细信息，可以在参考文献 [100] 中找到。



梗　　概

统计学习理论（SLT）声明了能成功从实例中进行学习的条件；也就是说，对于相同底层概率分布产生的新实例，训练数据的积极成果能转换成有效的泛化。**分布的稳定性**是至关重要的：好的老师绝不会用一些例子来教育学生，却又用完全不同的另一些例子来考试。换句话说，实例必须代表问题。可学习性的条件意味着假设空间（我们用于学习的“可调参数的灵活机器”）必须足够强大，使其在训练实例上有不错的表现（经验风险小），但又不能过于强大，以至于只记住了实例，却没有提取问题的深层结构。这一灵活性是由 VC 维度量化的。

SLT 展示了从数据中学习的天堂是存在的，但是对于大多数实际的问题，它并不显示进入天堂大门的实际步骤，通过直觉和交叉验证选择适当的核和参数才是成功的关键。

深度学习和 MLP 的最新成果带来了新的希望，“特征工程”和内核选择步骤可以完全自动化。这一领域的研究尚未形成定论，仍有新技术的突破空间，以及创造力引领下的特立独行、野蛮生长的精神。

第 12 章　最小二乘法和健壮内核机器

科学或许就是系统简化的艺术。

—— 卡尔　波普尔



支持向量机最初是基于这样的设想：将数据映射到高维空间，并在该空间中构造一个最优的分割超平面，即最大化“安全”间隔。就像 11.1.1 节一样，为了使数据点安全地正确地落在超平面的两侧，有如下**不等式**：

$$y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i+b)\geqslant 1 \qquad i=1,\cdots,\ell$$

再通过添加违反约束 ξ_i 修正为：

$$y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i+b)\geqslant 1-\xi_i \qquad i=1,\cdots,\ell$$

最大化间隔就是最小化 $\|\boldsymbol{w}\|$。对偶凸**二次规划**（QP）可以得到最优值，就像 MLP 和其他技术一样，不会收敛到一个局部极小值。

当研究者都追随着 SVM/凸二次规划的热潮时，有两个问题却未引起关注。第一个问题是如何选择适当的核。具有良好泛化能力的线性可分器需要恰当地度量训练样本及测试样本的相似性：

$$K(\boldsymbol{x},\boldsymbol{x}_i)=\boldsymbol{\varphi}(\boldsymbol{x})\cdot\boldsymbol{\varphi}(\boldsymbol{x}_i)$$

举个简单的例子，温和的老师会解决一个问题最主要的部分，然后将不重要的部分（使用二次规划找出最优超平面）留给学生。深度学习（10.1 节）就采用这种方式，它直接从数据中自动构建了许多中间的特征，以供后续的机器学习算法进一步学习。

第二个问题是计算效率。二次规划是可解的，但是在求解许多大规模问题时，CPU 所花费的时间会迅速增长。采用二次规划是因为存在不等式的约束，所以尝试舍弃不等式约束而得到更简单的等式约束是值得的。使用等式约束并对误差进行平方形式的罚分，将能得到类似之前良好的线性等式，可以更快地解决问题，也便于理解。本章将介绍**最小二乘支持向量机**领域的最新发展。我们将会看到，使用一些其他的方法，将使得二次罚分不会导致参数的**稀疏性**。除此之外，当样本存在**离群值**时，二次罚分会变得很脆弱，因为离群值巨大的偏差将会被平方。离群值可能是测量误差导致的，研究者通常会避免一些损坏模型性能的离群值，一个可行方法就是采用健壮的版本，即限制偏离误差罚分的大小，以防止其过大。

章首图中的弹簧可看作数据点和拟合模型之间弹簧形连接的物理解释，这种连接都具有二次势能。

12.1　最小二乘支持向量机分类器

继参考文献 [93] 在支持向量机中为函数估计引入岭回归后，Suykens 和 Vandewalle[106] 提出了基于核方法的最小二乘支持向量机分类器。

SVM 分类器的最小二乘变体可以通过改写 SVM 中的最小化问题得到：

$$\begin{aligned}&\text{最小化}_{\boldsymbol{w},b,e} \quad J_2(\boldsymbol{w},e)=\frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}+\frac{\gamma}{2}\sum_{i=1}^{\ell}e_{c,i}^2\\&\text{使服从} \qquad y_i\left[w^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_i)+b\right]=1-e_{c,i}, \quad i=1,\cdots,\ell\end{aligned}$$

可以通过调整超参数 γ，确定正则化项和二次误差之和的适当比例。上述的最小二乘 SVM（LS-SVM）分类器隐式地对应着一个二值目标 $y_i=\pm1$ 的回归模型。

使用 $y_i^2=1$，我们有：

$$\sum_{i=1}^{\ell}e_{c,i}^2=\sum_{i=1}^{\ell}(y_ie_{c,i})^2=\sum_{i=1}^{\ell}\left[y_i-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_i)+b)\right]^2=\sum_{i=1}^{\ell}e_i^2$$

这里 $e_i=y_i-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_i)+b)$，这个误差对于最小二乘数据拟合也是有意义的，所以最小二乘 SVM 和其对应的回归模型拥有相同的最终结果。注意误损函数 J_2 是由拟合误差的均方和（SSE）和一个对过大参数罚分的正则化项组成，这是一种训练多层感知机的标准方法，也和 4.7 节的岭回归有关。

可以通过构造如下拉格朗日函数来求解 LS-SVM 的回归量：

$$\begin{aligned}L_2(\boldsymbol{w},b,e,\boldsymbol{\alpha})&=J_2(\boldsymbol{w},e)-\sum_{i=1}^{\ell}\alpha_i\Big\{\left[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_i)+b\right]+e_i-y_i\Big\}\\&=\frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}+\frac{\gamma}{2}\sum_{i=1}^{\ell}e_i^2-\sum_{i=1}^{\ell}\alpha_i\Big\{\left[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_i)+b\right]+e_i-y_i\Big\}\end{aligned}$$

这里的 $\boldsymbol{\alpha}=(\alpha_1,\cdots,\alpha_\ell)^{\mathrm{T}}\in\mathbb{R}^\ell$ 是拉格朗日乘子，也被称作支持值。

通常，为了避免直接求解二次规划问题，使用拉格朗日乘子法，由目标函数极小值处的梯度等于 0，将会得到一个线性方程组（二次形式函数的导数是线性的）：

$$\begin{pmatrix} 0 & \mathbf{1}_\ell^{\mathrm{T}} \\ \mathbf{1}_\ell & \Omega+\gamma^{-1}I_\ell \end{pmatrix}\begin{pmatrix} b \\ \boldsymbol{\alpha} \end{pmatrix}=\begin{pmatrix} 0 \\ \boldsymbol{y} \end{pmatrix} \tag{12.1}$$

其中 $\boldsymbol{y}=(y_1,\cdots,y_\ell)^{\mathrm{T}}$，$\mathbf{1}_\ell=(1,\cdots,1)^{\mathrm{T}}$，$I_\ell$ 是 $\ell\times\ell$ 的单位矩阵，$\Omega\in\mathbb{R}^{\ell\times\ell}$ 是由 $\Omega_{ij}=\boldsymbol{\varphi}(\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_j)=K(\boldsymbol{x}_i,\boldsymbol{x}_j)$ 定义的核矩阵。

使用“核方法”，不必明确求出映射 $\boldsymbol{\varphi}$，只需求出内积即可。这种方法十分有用，因为权重向量 $\boldsymbol{w}$ 可以是无限维，在一些情况下我们几乎不可能求出它的映射。

通过求解线性方程组 (12.1) 而不是求解二次规划，我们就能得到一个分类器，作为函数预测，LS-SVM 的结果为

$$y(\boldsymbol{x})=\sum_{k=1}^{\ell}\alpha_k K(\boldsymbol{x},\boldsymbol{x}_k)+b$$

例如，径向基核（RBF）由宽度参数 σ 定义：

$$K(\boldsymbol{x}_1,\boldsymbol{x}_2)=\mathrm{e}^{-\frac{\|\boldsymbol{x}_1-\boldsymbol{x}_2\|^2}{\sigma^2}}$$

这种情况下，支持值 $\alpha_k=\gamma e_k$ 与数据点的误差是成比例的，而在标准 SVM 中，这些支持值大部分为 0。

一个学习双螺旋基准问题的 SVM 如图 12-1 所示。

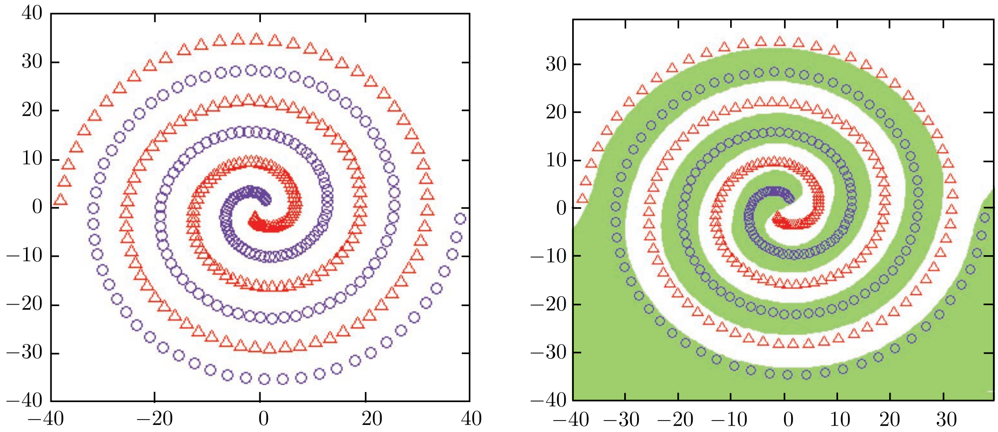


图 12-1 SVM 学习双螺旋的分类问题，两个类分别用圆圈和三角形表示。这幅图展示了 SVM 卓越的泛化性能

12.2 健壮加权最小二乘支持向量机

参考文献 [105] 中讨论了 LS-SVM 的健壮性和稀疏逼近问题。LS-SVM 得到的线性系统 [式 (12.1)] 可以通过直接求解或类似于共轭梯度法的逐步迭代（21.3.2 节）有效地解决。但是，LS-SVM 的求解有一些潜在的缺点。第一点不足就是**缺少稀疏性**，所有的数据点都影响着整个模型，数据点的相对重要性通过其支持向量给出。第二点不足广为人知，是使用的均方和损失函数没有正则化，这将会导致当数据中存在**离群值**，或者当误差变量服从高斯分布这一假设不成立时，估计函数**缺少一定的健壮性**。

离群值的问题是较大误差的平方导致误差过大，可以通过**加权的最小二乘法**，即对那些非常大的误差进行加权来解决，这将会使估计更稳健。

这种方法首先应用非加权的 LS-SVM 计算结果，然后根据第一步误差变量的结果对 LS-SVM 的误差变量加权，最终需要在非加权的 LS-SVM 上解决一系列的加权 LS-SVM 问题。这种做法是为了**根据训练数据调整所用的基本损失函数**，而不是直接使用事先定义好的损失函数。

为了在之前 LS-SVM 的结果上得到一个稳健估计，在接下来的步骤中，对误差变量 $e_k = \alpha_k/\gamma$ 加以权重因子 v_k，将得到以下优化问题：

$$\min_{\boldsymbol{w}^*, b^*, e^*} \frac{1}{2}\boldsymbol{w}^{*T}\boldsymbol{w}^* + \frac{1}{2}\gamma\sum_{k=1}^{\ell} v_k e_k^{*2}$$

加权 LS-SVM 问题的未知变量用 * 符号标出。

权重 v_k 的选择主要基于（非加权）LS-SVM 中的误差变量 $e_k = \alpha_k/\gamma$。首先得到 $\hat{s}$，然后就可以得到 LS-SVM 中误差变量 e_k 标准求导的一个稳健估计：

$$\hat{s} = \frac{\mathrm{IQR}}{2 \cdot 0.6745} \tag{12.2}$$

四分位差（IQR）为较大的四分位数与较小的四分位数之差，显然，类似于离群值的极端数据不在该估计范围内，所以这种方法是健壮的。具体来说，稳健估计 [91] 可以通过计算下式得到：

$$v_k = \begin{cases} 1 & |e_k/\hat{s}| < c_1 \\ \dfrac{c_2 - |e_k/\hat{s}|}{c_2 - c_1} & c_1 \leqslant |e_k/\hat{s}| \leqslant c_2 \\ 10^{-4} & \text{其他情况} \end{cases} \tag{12.3}$$

常数 c_1 和 c_2 通常取 $c_1 = 2.5$ 和 $c_2 = 3$，这是考虑到高斯分布中，残差的值很少有超过 $2.5\hat{s}$ 的，所以那些在高斯分布中过大的误差变量将会被赋予越来越小的权重。

如果需要，上述过程可以迭代重复进行，不过在实际应用中，一次额外的加权 LS-SVM 就足够了。最后的算法请参考图 12-2。

LS_SVM 算法 (ℓ 个训练数据点)

- 使用式 (12.1) 中的线性方程组, 通过 K 折交叉验证找到最优 (r, σ)
- $e_k \leftarrow \alpha_k / \gamma$
- 使用式 (12.2) 计算 e_k 分布的 $\hat{s}$ 值
- 通过计算式 (12.3) 确定基于 e_k 和 $\hat{s}$ 的权重 v_k
- 给定以下模型, 根据式 (12.1) 求解 α^* 和 b^*
- $y(\boldsymbol{x}) = \sum_{k=1}^{\ell} \alpha_k^* K(\boldsymbol{x}, \boldsymbol{x}_k) + b^*$

图 12-2 加权 LS-SVM 算法

估计量的崩溃点是稳健估计的一个重要概念。崩溃点表示使估计量崩溃的最小比例，即当给定数据集中最少有百分之多少的数据被（离群值）污染时，能使最终得到的估计量与原始数据得到的估计量相差任意远。在线性回归中，标准不加正则项的最小二乘估计崩溃点很低，使用加权 LS-SVM 可以大幅提高崩溃点的值。

12.3 通过修剪恢复稀疏

标准 SVM 具有稀疏性，因为许多 α_k 的值为 0，而由于在最优情况下 $\alpha_k = \gamma e_k$，LS-SVM 则没有这样的性质。支持值揭示了数据点对模型贡献的相对重要性。

就像多层感知器可以根据黑塞矩阵进行剪枝 [例如*最佳大脑破坏法*（optimal brain damage）[77] 和*最佳大脑手术*（optimal brain surgeon）[54]]，参考文献 [105] 提出了根据解向量自身对 LS-SVM 进行剪枝。通过对排序好的支持值谱系逐步剪枝，即将较小的 α_i 归零，就能将稀疏性强加至加权 LS-SVM 上：通过这种方法，不重要的数据点（根据其支持值）将被舍去，LS-SVM 在剩下的数据点上重新计算，但需要在整个训练数据集中验证。

通过舍去相对少量的最无意义的数据点（设其 α_k 值为 0），并重新计算 LS-SVM，就能得到一个稀疏的最优结果。为了保证良好的泛化能力，在每一步剪枝时都可以最优化 (γ, σ)，比如通过定义一个独立的验证集或 10 折交叉验证。图 12-3 描述最终的算法流程。

LS_SVM_pruning 算法 (ℓ 个训练数据点)

- $\ell' \leftarrow \ell$
- 性能下降前一直重复
  - 对ℓ'训练数据应用 LS_SVM 算法
  - 根据下降梯度 $|\alpha_k^*|$ 分类训练数据
  - 在分类的 $|\alpha_k^*|$ 范围内移除最后 M 个数据点
  - $\ell' \leftarrow \ell' - M$

图 12-3 加权 LS-SVM 剪枝算法

通常，在对 LS-SVM 进行剪枝时，可以不改变 (γ, σ) 的值，等到模型的泛化能力开始退化时（例如通过验证集或交叉验证的平均值来检查 [109]），再去更新 (γ, σ) 的值。相比于其

他方法需要解决一个包含很多超参数选择的二次规划问题，这种方法的一个潜在的优势就是 (γ, σ) 的计算可以在局部进行。

12.4　算法改进：调谐 QP、原始版本、无补偿

对于 SVM 的改进主要涉及两个方面，一个是适应于 SVM 的二次规划的详细实现，另一个是对问题定义的细微修改，这些改进对于 SVM 的 CPU 运行时间和最终性能都有潜在的巨大影响 [70]。

式 (11.7) 中的二次形式包含一个矩阵，它的元素个数是训练样本数的平方（矩阵元素包含了每两个样本间所有可能的核“相似度”）。参考文献 [27] 首先提出将大型 SVM 学习问题分割成一系列较小的优化任务的方法，即**分块法**（chunking algorithm）。这种方法首先从训练集中随机取出一个子集，在该数据集上解决 SVM 问题，然后不断迭代添加那些不满足最优条件的样本。

参考文献 [87] 中的工作展示了相比于使用现成的二次规划软件，使用专门设计的二次规划求解算法的效率能提升多少（以及，研究求解的数学细节，效率又能提升多少）。**序列最小优化算法**（Sequential Minimal Optimization，SMO）将求解的大规模二次规划问题分割成一系列较小的二次规划问题。这些较小二次规划问题可以直接解析求解，从而避免了二次优化耗时的数值计算，也就是内循环。SMO 对内存的需求随着训练数据集的大小线性增长，所以 SMO 可以处理非常大的训练集。由于避免了大规模矩阵的计算，SMO 的规模随着训练数据的增长速度而介于线性和二次方之间，而标准的投影共轭梯度（projected conjugate gradient，PCG）分块法的规模随着训练数据的增长速度而介于线性和三次方之间。SMO 的时间复杂度主要是由所求的 SVM 决定的，因此 SMO 计算线性 SVM 和稀疏的数据集的速度十分快。

参考文献 [70] 提出了一种针对大规模问题的 SVM 训练过程的优化算法（SVMlight）。该算法基于一种分解策略，通过一种快速高效的方式解决了 SVM 中工作集中参数的选择问题。具体来说，该算法引入了一种在优化阶段缩小问题规模的方法：在 SVM 优化阶段，可以很早确定某些样本不太可能成为支持向量（support vector，SV），因此可以通过排除这些样本来缩小问题规模。当一个 SVM 的数据集中支持向量占整个样本集的比例很小时，这种算法特别有效。同时，SVMlight 的内存需求随着训练数据和支持向量的数量线性增长。

参考文献 [37] 提出了**原始空间中解决 SVM** 的方法。大部分关于 SVM 的文献都关注其对偶优化问题。参考文献 [37] 的作者认为 SVM 的原始问题也可以得到有效解决，并且研究者没有理由忽视这个可行的方法。另一方面，从原始空间的视角来看，可以研究一些新的大规模 SVM 的训练算法家族。通常，使用对偶问题来解决 SVM 的主要原因是：

(1) 对偶理论可以很方便地处理约束条件；

(2) SVM 的对偶优化问题可以写成点积的形式，因此可以使用核函数方法。

SVM 原问题的牛顿优化法有着和对偶优化法相同的计算复杂度，但涉及近似解时，原优化方

法更胜一筹，因为它更关注于我们想要最小化的函数：原始目标函数。原优化方法在大规模最优化问题上或许更具优势。显然，当训练数据的数量很多时，支持向量的数量也会很多，想要得到问题确切的解将变得很困难，所以一般需采取近似的方法，但是在对偶空间中使用近似的方法显然是不明智的，因为**对偶空间得到的近似值并不能保证其在原空间中也是一个好的近似**。

另一方面，参考文献 [102] 对**不带偏移量的支持向量机分类器**构建并分析了一种训练算法。过去，SVM 是基于特征空间中线性判别面的几何形式设计的，如图 11-3 所示，这种形式自然使用偏移量 b，即判别面相距原点的偏移，但是这种几何形式有着严重的弊端。尽管这种形式具有很好的可视性，但我们**绝对不能简单地根据其在低维空间的示例去选择算法**。

结果表明，解决带偏移量的 SVM 最优化问题比不带偏移量有着更多的约束。偏移量导致对偶优化问题中多了一个等式约束，该等式使得 SVM 的一些常用解法，如 SMO，必须在每次迭代中更新至少两个对偶变量的值 [87]。

参考文献 [102] 的作者针对不带偏移量的 SVM 构建了一些算法。这些算法不仅比那些带偏移量的 SVM 更准确，还运行得更快。



梗 概

最小二乘支持向量机采用等式而非不等式进行分类（通常将正例映射到 +1，负例映射到 −1），这样，对于误差的二次罚分经过偏导并令梯度为 0 后，将得到一个**线性方程组**。

非常大的偏差会导致二次罚分快速增长，因此很少的**离群值**就能导致模型失灵。使用**稳健统计**的方法，将离群值对收益函数的影响降到最低，可以消除其对模型过度干扰，即通过给那些非常大的误差赋予很小的权重，得到**健壮加权最小二乘 SVM**。

二次表达式中稀疏性的缺失可以通过**剪枝**的方法恢复，那些几乎无意义的数据点将被移除，LS-SVM 在剩余的数据集上重新计算。

传统的最小二乘法最小化残差的均方和，仍然能给予一些新型方法（比如 SVM）强有力的支持，所以当与新型方法比较的时候，绝对不要低估优秀的传统方法和线性代数方法。

第 13 章　机器学习中的民主

每个共和国都有两个相互冲突的阶层：平民和贵族。正是在这样的冲突中，诞生了捍卫自由的法律。

—— 马基雅弗利



你已经发现，用于解决监督学习中的问题的有效技术有许多，每项技术的区别在于模型选择和元参数的不同：当考虑到这样的灵活性时，人们很容易就能想到**许多可用于完成给定任务的模型**。

当面对这样丰富的选项时，人们可以只选择最好的模型（以及最佳的元参数）并扔掉其他一切，或者永远不嫌好东西多并尝试所有的可能性——至少是最好的那些。你已经耗费精力和 CPU 时间来选择最佳模型和元参数，并顺带着生成许多模型。是否有合理的方法来回收它们，使得之前的努力不会白费？放轻松，本章不再引入全新的模型，而会**用灵活、创新和有效的方法来处理许多不同的模型**。在某些情况下，这样做的优点是很明确的，使用多种模型与否，可以决定 ML 竞赛的输赢。

本书的中心思想是，许多 ML 原理可以类比于人类的某种学习形式。询问**专家团体**是一个人做出重要决定的方式，如果参与者有不同的专长和与专业水平相当的敬业精神，那么这个团体就能工作得很好。背景、文化、性别上的差异被认为是创新型商业成功的重要因素。民主本身可以被认为是一种凝聚公民知识以做出可执行决策的务实方式（好吧，也许并不总是最优的，但总比一个人说了算好）。

在 6.2 节中我们已经遇到了一个有创意的用法，将很多分类树当作**分类森林**。本章会总结更多有效利用不同 ML 模型的主要技术，重点讨论架构原则，也会涉及一些基本的数学原理。

13.1 堆叠和融合

如果你参加了一个机器学习竞赛（或者如果你想赢得一个合同，或者需要为某个迫切的业务寻求一个解决方案），没准你会尝试不同的方法，并拿出一大堆模型。就像好的咖啡，融合起来可能会更加美味。

有两种简单的方式来组合各种模型的输出：通过**投票**和通过**平均**。假设我们的任务是将模式分为两类。在投票中，每个训练得到的模型为某一类投票，收集选票后，最终的输出是拥有更多选票的那一个，就像少数服从**多数**的基本民主程序一样。如果每个模型正确分类的概率大于 1/2，**并且不同模型的误差是不相关的，那么，随着模型数量的增长，**M **个模型中大多数出错的概率将为零**。然而，在实际情况中，误差之间往往是相关的。如果一个模式难以识别，那么对于许多模型来说也将如此，其中多个模型出错的概率会高于它们单独错误概率的乘积，优势就不会那么显著。想一想信封上的邮政编码，如果数字上有污渍，就会令许多模型难以识别，从而将误差关联起来。

若任务是预测概率（给定输入模式所对应类别的后验概率），则另一种选择是求各个概率的平均值。顺便说一句，对实验测量值求平均是**方差缩减**的标准方法。在一定的条件下，统计学中的“大数定律”解释了为什么大量尝试的结果的平均值往往接近于期望值，并且解释了为什么尝试次数越多就越接近。

虽然看上去很简单，投票和平均有个共同的缺点：它们一视同仁地对待所有模型，因此最优模型的性能会埋没在一团平庸的模型之中。一个决策越复杂，就越需要针对不同**专家进行加权，并且加权应该依赖于特定的输入**。

你其实已经有了一把解决专家加权问题的利器：机器学习本身！只需在其上添加一个线性模型，由 0 级模型（专家）连接不同的输出，并让 ML 确定最佳的权重（见图 13-1）。这是**堆叠泛化**（stacked generalization）[117] 的基本思路。请注意，为了避免过度训练，用于训练堆叠模型（确定顶部附加层的权重）的训练实例必须从未在之前任何模型的训练中使用过。训练实例就像鱼肉：如果放了太久，那么就会变臭！

堆叠泛化有如下的结果 [108]。

- 如果可行，**使用类别的概率**作为原来的 0 级模型输出（而不是类别的预测）。概率的估计会提供一些置信度的信息，而不仅仅是预测。保持它们的原样，会以更高的水平给出更多的信息。
- 通过向优化任务添加约束来确保组合**权重非负**。它们对堆叠回归准确率的提高而言是必需的，对于分类任务而言则不是必需的，但在这两种情况下，它们都提高了 1 级模型的**可解释性**（零权重是指不使用相应的 0 级模型，权重越高，模型就越重要）。

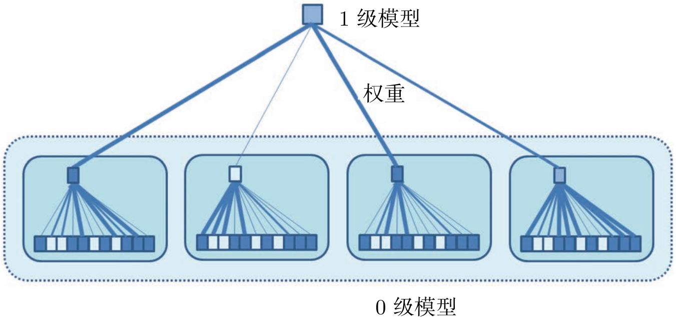


图 13-1　通过在模型之上增加附加模型来融合不同模型（堆叠）

如果你的胃口还没有满足，可以用多层次组合来进行实验，或者使用更多结构化组合。例如，你可以在 MLP 和决策森林之上进行堆叠，或者通过添加另一层次（见图 13-2）结合已经完成的一组堆叠模型。管理的模型越多，对上面提到的“发臭的实例”规则就必须更加小心。高层次的模型并不需要是线性的：这会损失一些可解释性，但非线性模型可能会使最终的结果更好。

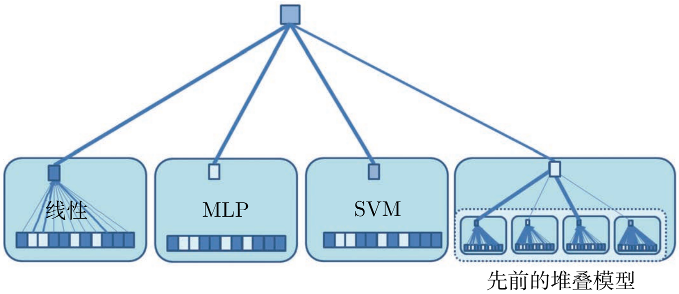


图 13-2　堆叠可应用于不同的模型，其中包括先前已经堆叠的

特征加权线性堆叠是一个有趣的选择 [98]。作为一个具体的例子，它展示了现实世界的应用如何产生优雅的解决方案。某些情况下，我们会有一些附加信息，即原始输入特征之外的“**元特征**”。例如，如果目标是预测客户偏好并为其提供各种产品（在协同过滤和建议的场景中），模型的可靠性可以根据附加信息而变化。举例而言，对于评价了许多产品的用户，某模型 A 可能更可靠（在这种情况下，该用户评价过的产品数量是“元特征”）。为了保持线性回归，同时又允许权重依赖于元特征（对于评价了更多产品的客户，模型 A 可以有更大的权重），可以要求权重与元特征的关系是线性的。如果 $g_i(\boldsymbol{x})$ 是对于 0 级模型 i 的输出，$f_j(\boldsymbol{x})$ 是第 j 个元特性，那么权重为：

$$w_i(\boldsymbol{x}) = \sum_j v_{ij} f_j(\boldsymbol{x})$$

其中 v_{ij} 是由堆叠模型学习的参数。1 级输出是：

$$b(\boldsymbol{x}) = \sum_{i,j} v_{ij} f_j(\boldsymbol{x}) g_i(\boldsymbol{x})$$

这就产生了以下**特征加权线性堆叠**的问题：

$$\min_{(v_{ij})} \sum_{\boldsymbol{x}} \sum_{i,j} \left(v_{ij} f_j(\boldsymbol{x}) g_i(\boldsymbol{x}) - y(\boldsymbol{x})\right)^2$$

由于该模型仍然与 v 呈线性关系，我们可以使用标准的线性回归分析来确定最佳的 v。像往常一样，永远不要低估线性回归的能力，前提是能以适当的方式来创造性地运用它。

13.2 实例操作带来的多样性：装袋法和提升法

为了成功地建立一个民主的 ML 系统，人们需要**一套准确且多样化的分类器**或回归器，也称为**集成**（ensemble），像一群在一起表演的音乐家。在文献中，**集成方法**是这些技术的传统术语，其近义词是**多分类器系统**（multi-classifiers system）。

多种技术可以根据其创建多样性的主要方式组织起来 [43]。

用**训练样本的不同子集**来进行训练是一种可行的方式。在**装袋法**[bagging，即“自助汇合”（bootstrap aggregation）] 中，不同的子集由带放回的随机抽样产生（同一个实例可以被多次抽取）。每个自助副本包含原始实例约三分之二（实际上 $\approx 63.2\%$）。不同模型的结果通过求平均，或者根据多数决定原则汇合起来。如果数据有微小变化，不稳定的学习算法常常会使结果变化很大，装袋法在提高稳定性方面表现良好。如 6.2 节所述，装袋法可以从一组分类树中产生**分类森林**。

交叉验证团体（cross-validation committee）将训练数据划分的不相交子集用来准备不同的训练集。这种情况下使用交叉验证推定模型性能，各种模型的产生是副产品（并且不需要耗费额外的 CPU）。

操纵训练集的更动态的方式是通过**提升法**（boosting）。这一术语是基于这样一个事实，即弱分类器（尽管性能必须比随机分类器略好）的性能可以被“提升”，从而得到一个精确的团体 [45]。就像装袋法和自助法一样，提升法也创建多个模型，但每次迭代产生的模型是建立在**自适应**的方式上的，能直接改进之前创建的模型的组合。算法AdaBoost 维护一组训练实例的权重。每次迭代后，权重会更新，当前模型**分类错误的实例会有更高的权重**（见图 13-3）。就像一位专业的教师，他在课程中会更愿意使用那些还未被学生完全理解的例子。

最终的分类器 $h_f(\boldsymbol{x})$ 由各个分类器加权投票来决定。每个分类器的权重反映了它在这些加权训练集上训练的准确率：

$$h_f(\boldsymbol{x}) = \sum_l w_l h_l(\boldsymbol{x})$$

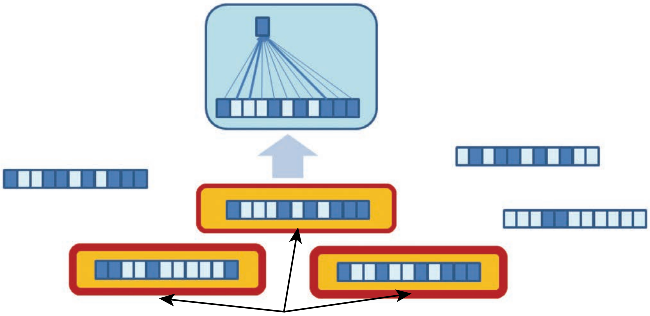


图 13-3　在提升法中，当训练新的模型添加到团体中时，当前迭代中分类错误的实例会被赋予更大的权重

由于我们笃信优化的强大力量，理解提升法的最好方法就是通过它所优化的函数。通过改变优化的函数或者通过改变优化的细节，可以得到（和理解）不同的变形。为了定义误差函数，假设每个训练实例的输出 y_i 只能是 $+1$ 或 -1。值 $m_i = y_i h(\boldsymbol{x}_i)$ 称为分类器 h 在训练数据集上的间隔，分类准确时为正，否则为负。正如 13.6 节中将要看到的那样，适应提升法 AdaBoost 可以被看作最小化下面的误差函数的**阶段性算法**：

$$\sum_i \exp\left(-y_i \sum_l w_l h_l(\boldsymbol{x}_i)\right) \tag{13.1}$$

该目标函数是加权分类器的间隔的负指数。这其实等价于**最大化训练数据集上的间隔**。

13.3　特征操作带来的多样性

特征的不同子集可以用来训练不同的模型（见图 13-4）。某些情况下，根据不同的性质来组合特征可能是有用的。在参考文献 [38] 中，这一方法用来识别金星上的火山，并表现出了媲美人类专家的性能。由于不同的模型都需要足够精确，使用特征子集只有在特征高度冗余的情况下才会有效。

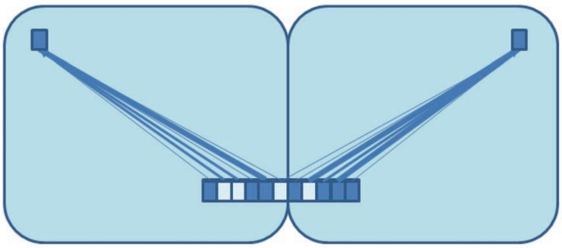


图 13-4　使用特征的不同子集来创建不同的模型。该方法不局限于线性模型

13.4 输出值操作带来的多样性：纠错码

纠错码（ECC）的设计是为了在有噪线路传输时，即使有一定数量的错误，也能保证健壮性（见图 13-5）。例如，如果“一”的码字是“111”,“零”的码字是“000”，那么一个二进制位的错误如“101”是可以接受的（该码字将被映射到正确的“111”）。参考文献 [44] 中提出的**纠错输出码**可以用来设计分类器团体。

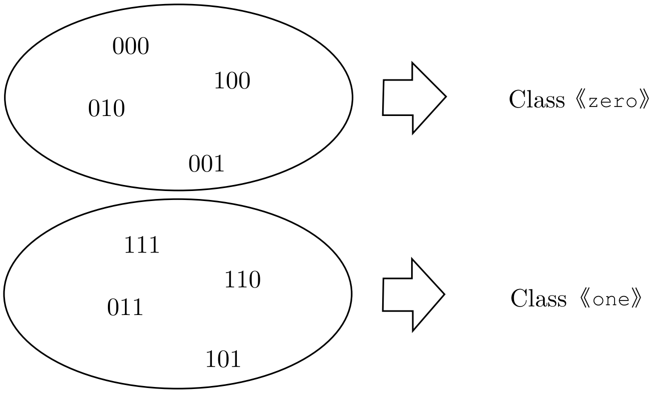


图 13-5　纠错码里设计了冗余编码来抵抗一定数量的错误二进制位（bit）

用 ECC 来设计专家团体，是将每个输出分类 j 用一个 L 位的码字 C_j 来编码。团体中训练得到的第 l 个分类器用于预测码字的第 l 位。经由团体中 L 个分类器生成所有的位之后，输出与其最相近码字对应的类别（用汉明距离，即不同位的数量）。由于码字是冗余的，个别分类器造成的一定量的误差可以被整个团体纠正。

当然，不同集成方法也可以结合使用。例如，纠错输出码可以和提升法相结合，当然也可以与特征选择相结合，某些情况下这些结合有非常好的结果。

13.5 训练阶段随机性带来的多样性

许多训练算法本身就带有随机性。这种随机性是得到多样化模型的一个很自然的方法（通过改变随机数生成器的种子）。例如，MLP 从随机初始权重开始。树算法中，在确定内部节点要测试的特征时，可以用随机的方式，就像得到决策森林那样。

最后，大多数用于训练的优化方法也有加入随机性的空间。例如，随机梯度下降方法以随机的顺序来展现模式。

13.6 加性logistic回归

前文提到了提升法，它按次序在重新加权的训练实例上应用分类算法，然后采纳这些模型

输出的加权多数票。

作为优化的强大力量的例子，提升法（boosting）可以理解为一种应用**加性 logistic 回归的方法，这是一种以前推阶段性方式来拟合一个加性模型** $\sum_m h_m(\boldsymbol{x})$ **的方法**[46]。

接下来从简单的函数开始：

$$h_m(\boldsymbol{x}) = \beta_m b(\boldsymbol{x}; \boldsymbol{\gamma}_m)$$

它由一组参数 $\boldsymbol{\gamma}_m$ 和作为权重的乘数 β_m 确定。可以将 M 个这样的函数组合成一个加性模型：

$$H_M(\boldsymbol{x}) = \sum_{m=1}^{M} h_m(\boldsymbol{x}) = \sum_{m=1}^{M} \beta_m b(\boldsymbol{x}; \boldsymbol{\gamma}_m)$$

使用**贪心逐步向前法**（greedy forward stepwise approach），我们可以确定每次迭代中最优的参数 $(\beta_m, \boldsymbol{\gamma}_m)$，这样**新加进来的简单函数往往能纠正前面模型** $F_{m-1}(\boldsymbol{x})$ **的误差**（见图 13-6）。如果将最小二乘用作拟合判据：

$$(\beta_m, \boldsymbol{\gamma}_m) = \underset{(\beta,\boldsymbol{\gamma})}{\operatorname{argmin}}\, E\left[\left(y - F_{m-1}(\boldsymbol{x}) - \beta b(\boldsymbol{x}; \boldsymbol{\gamma})\right)^2\right] \tag{13.2}$$

其中 $E[\cdot]$ 是期望值，由所有实例的和估计得到。这一贪心式的过程可以一般化为**向后拟合法**（backfitting），每次迭代都拟合一个参数对 $(\beta_m, \boldsymbol{\gamma}_m)$，并不一定是最后一个参数对。注意这种方法仅仅要求算法对数据拟合单个弱学习器 $\beta b(\boldsymbol{x}; \boldsymbol{\gamma})$。这里的数据是在原始数据基础上反复修改过的（见图 13-7）：

$$y_m \leftarrow y - \sum_{k \neq m} h_k(\boldsymbol{x})$$

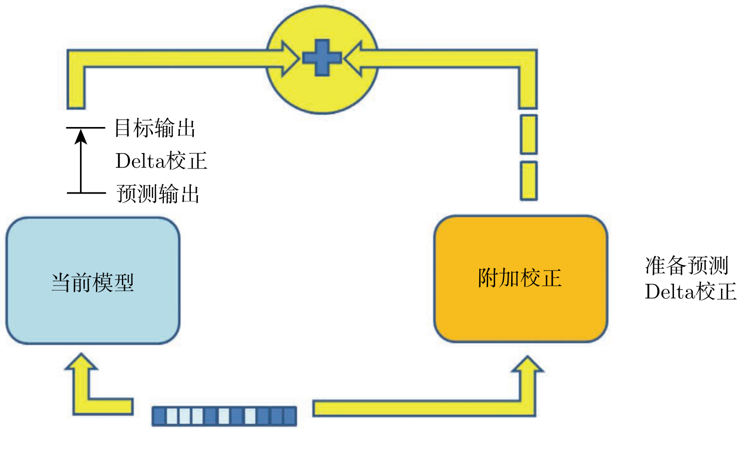


图 13-6　加性模型步骤：在训练实例上测量当前模型的误差。加入第二个模型是为了抵消这一误差

对于分类问题，用平方误差损失（关于理想的输出值 0 或 1）会导致麻烦。如果想要估计后验概率 $\Pr(y=j|\boldsymbol{x})$，将不保证该输出被限制在 $[0,1]$ 的范围内。另外，误差平方不仅仅惩罚真实的误差（比如应该是 1，而预测是 0），而且也**制裁分类“过于正确”的**（比如应该是 1 而预测是 2）。

logistic 回归能解决这些问题（见 8.1 节）：采用加性模型 $H(\boldsymbol{x})$ 来预测一个“中间”值，然后用 logistic 函数将其挤压（squash）到正确的 $[0,1]$ 区间内，以获得最终的概率形式的输出。

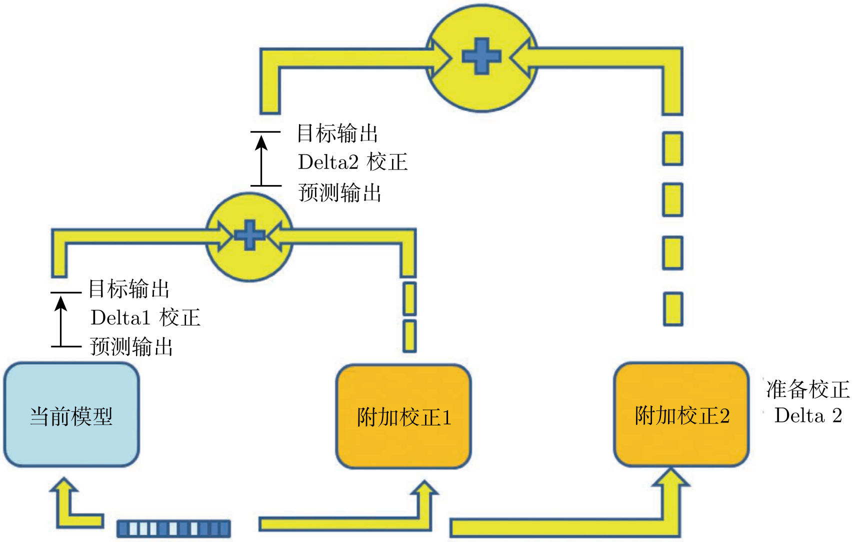


图 13-7 加性模型中的贪心逐步向前法，通过添加新的组件的迭代来取消剩余的误差

加性 logistic 模型的形式是：

$$\ln\frac{\Pr(y=1|\boldsymbol{x})}{\Pr(y=-1|\boldsymbol{x})}=H(\boldsymbol{x})$$

其中左边的 分对数（logit）变换将概率值 $\Pr(y=1|\boldsymbol{x})\in[0,1]$ 单调地映射到整个实数轴上。因此，分对数变换（及其逆变换）

$$\Pr(y=1|\boldsymbol{x})=\frac{\mathrm{e}^{H(\boldsymbol{x})}}{1+\mathrm{e}^{H(\boldsymbol{x})}} \tag{13.3}$$

保证了概率估计落在正确的 $[0,1]$ 区间内。事实上，$H(\boldsymbol{x})$ 为式 (13.3) 中的 logistic 函数的输入建模。

现在，如果考虑期望值 $E\left[\mathrm{e}^{-yH(\boldsymbol{x})}\right]$，可以证明它在满足以下条件的时候是最小的：

$$H(x)=\frac{1}{2}\ln\frac{\Pr(y=1|\boldsymbol{x})}{\Pr(y=-1|\boldsymbol{x})}$$

即 $\Pr(y=1|\boldsymbol{x})$ 的对称 logistic 变换（注意前面的因子 1/2）。一个有趣的结果是，AdaBoost 通过类似牛顿法更新[①] 来最小化 $E\left[\mathrm{e}^{-yH(\boldsymbol{x})}\right]$ 从而建立加性 logistic 回归模型。技术细节和其他相关研究见参考文献 [46]。

13.7　民主有助于准确率–拒绝的折中

在模式识别系统的许多实际应用中，还有另一种“旋钮”需要打开：拒绝为某些实例分类。**对于一些难以处理的实例，拒绝对其分类，然后进行人工处理**（或者更复杂和昂贵的二级系统），要好过接受一切并对其分类。作为一个例子，在光学字符识别（例如邮政编码识别）中，由于书写不清楚，或者分割和预处理的错误，可能会出现难以识别的情况。在这种情况下，人工处理常常可以给出一个更好的分类，或者通过查看原来的明信片来修正预处理中的错误。假设 ML 系统有这样的一个附加旋钮，当它打开时，可以拒绝某些实例。这样就有了图 13-8 那样的**准确率–拒绝**（accuracy-rejection）**曲线**，它描述了可达到的准确率性能和拒绝率之间的函数关系。如果系统正在以智能的方式工作，**最难的未决实例将首先被拒绝**，因此，即使拒绝率很小，准确率也会快速增加。一个相关信号检测的“折中”曲线是接受者操作特征（ROC），图示表明了二元分类系统的性能，其鉴别阈值是变化的，通过绘制真阳性占实际阳性总数的比例（TPR= 真阳性率）和假阳性占实际阳性总数的比例（FOR = 假阳性率）在不同的阈值设置下的情形而来。

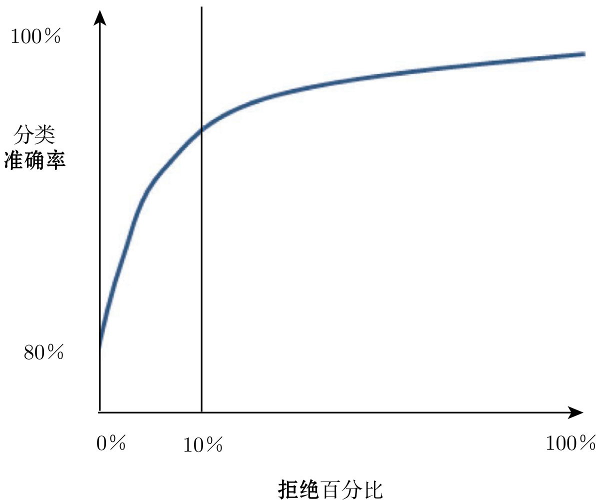


图 13-8　准确率–拒绝的折中曲线。可以通过拒绝一些疑难的实例来得到更高的精度

① 用牛顿式的步骤进行优化，接下来的章节会进一步解释，它意味着推导参数的二次近似，并且最优参数通过二次模型最小值得到。

为简单起见，下面来考虑一个二元分类的问题。在这一问题中，经过训练的模型输出类别为 1 的**后验概率**的近似值。如果输出接近 1，判定是明确的，这时正确的分类是 1 的概率很高。但如果输出接近 0.5 就会产生问题。在这种情况下，该系统“未判定”。如果所估计的概率接近于 0.5，这两个类别具有类似的概率，误差将会频繁出现（如果概率值是正确的，那么在这种情况下误差的概率等于 0.5）。如果正确的概率是已知的，当$P(类=1|x)$ 大于$1/2$，理论上最佳的**贝叶斯分类器**判定为类别 1，否则为类别 0。误差将等于剩余的概率。例如 $P(类=1|x)$ 为 0.8 时，误差将有 0.2 的概率（给定 x 是类别为 2 的实例，却被分为类别 1 的概率）。

为后验概率设定正阈值 T，要求它大于 $(1/2+T)$ 是最好的“旋钮”开关，通过拒绝不满足该标准的实例来增加精确度。一种情况是拒绝接近两个类之间的边界的实例，此时这两类实例按照概率接近 1/2（见图 13-9）的模式混合起来。

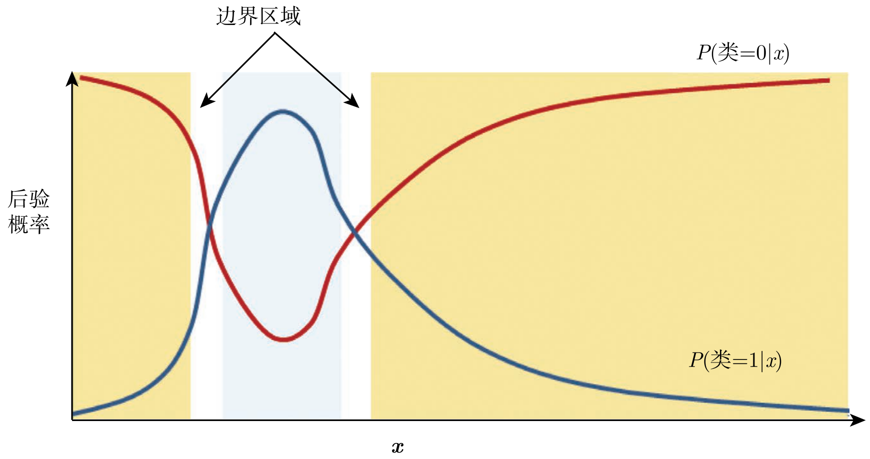


图 13-9 一个贝叶斯分类器中的过渡区域。如果拒绝落在接近边界的过度区域的实例，那么平均准确率将会上升

现在，如果概率是不知道的，但可以通过机器学习估计得到，那么拥有分类器团体就有了更多机会来获得更高的灵活性，以及实现更好的准确率–拒绝曲线 [19]。例如，通过激活每一个具有不同概率的分类器，可以得到**团队的概率组合**，或者多个分类器的组合。如果它们之中的全体或者**大多数**的意见是一致的，那就意味着拥有高置信度的实例，因此系统接受这些实例。最后，通过考虑输出概率（不仅是分类）、平均值和阈值化可以获得更高的灵活性。如果有两个以上的类，就可以要求平均概率高于第一阈值，而与第二个最佳分类的距离超过第二阈值。

实验结果表明，在解决任务时得到的许多分类器，如果能通过智能的方式进行重用，准确率–拒绝折中法很容易获得理想的结果和更高的灵活性。



梗　概

拥有一些**不同**但是**准确率**相近的机器学习模型，使得我们能够提升性能，从而超越单个系统（**如机器学习中的集成方法、团体方法、民主方法**）。

在**堆叠和融合**方法中，各种系统通过在单个模型的输出的顶端加入另一层而结合起来。

有多种不同方法可以在战略层面创造多样性。在**装袋法**（**自助汇合**）中，对同一组实例进行带放回的采样。**提升法**与加性模型相关，我们训练一系列模型，以确保当前系统中最难处理的实例会在最新添加的部分中获得**较大的权重**。使用不同的特征子集或者不同的随机数生成器也可能创造多样性。**纠错输出码**使用一组冗余的模型为各种输出位编码，以增强针对个别错误的健壮性。

加性 logistic 回归是一种优美的方式，它通过加性模型和牛顿式的优化方案来解释提升法。优化提升我们对提升法的理解。

机器学习中的集成方法就像爵士乐：整体大于部分之和。爵士乐手或模型在一起工作，互帮互助，依靠集体的力量比仅靠自己能创造更多。

第 14 章　递归神经网络和储备池计算

音乐是一个水池 ⋯⋯ 声音的水池。

—— 德克斯特　戈登



在神经网络和机器学习研究领域，一个“备受推崇的假说”长时间占据着统领地位——为了解决越来越复杂的学习问题，计算机需要竭尽所能，从越来越错综复杂的数据中提取出构造块（特征）。深度学习、监督预训练和阶段性特征提取都是遵循该思想的例子。

如果否定上述假设，就得到了**储备池学习**（reservoir learning）。储备池学习的思想是预先**在储备池中随机生成大量特征**，然后挖掘这些特征，建立最终的学习系统——通常采用极小二乘法来拟合储备池的隐藏输出和问题的实际输出。虽然储备池学习对于建立有效的学习系统看起来过于简单潦草，但是越来越多的实验证明，在许多情况下它的效率是极高的。在一些机器学习的案例中，储备池学习能生成比较理想的结果，或至少能快速提供一些初始的结果，这些结果可以使用额外的调参手段快速进行优化。

相比于机器学习复杂的训练机制，从**生物学的角度**可能更容易理解上述的储备池技术。例如，我们可以很快学会骑自行车、唱歌、说出新单词，实际上这些学习过程中只需很少的范例。这说明了**“随机”构造块**的有效性，我们从储备池中挖掘这些构造块，在合适的结构下能快速构造出想要的学习系统并加以微调。

14.1　递归神经网络

直到现在，我们在建立机器学习系统时并没有时间、历史、记忆的概念。换句话说，时间和迭代只存在于训练中，而在学习系统运行时并不加以考虑。系统运行时，输出只和输入有关，因为系统按照单次“前馈”的方式运行，不存在循环的部分。但生物的学习系统并非总是按照这样简单的方式运行。例如我们唱歌时，输出的歌声不仅仅依赖于当前输入的歌词和曲调，还与之前的输入和输出有关，演奏音乐、说话、心跳、呼吸、行走等也是如此。这些过程包含循环和振荡，是一种比单步输入–输出系统（仅仅实现了数学上的函数）更富有动态的行为。

人工**递归神经网络**（recurrent neural network，RNN）与广泛应用的前馈神经网络的区别在于，递归神经网络的拓扑结构中存在循环。在递归神经网络具有一定的记忆能力的前提下，它的输出有一些会反馈给网络中的点，继而影响网络之后的输出。依不同的模型而定，计算可以通过同步方式（类似于全局时钟使每个神经元按时收集数据，并根据当前输入生成输出）、异步方式（每个神经元被随机唤醒并更新输出）或者数学上微分方程持续动态调控的方式进行。图 14-1 展示了递归神经网络的基本结构：隐藏层神经元的输出可以作为输入反馈到其他神经元；输出结果可以反馈到隐藏层神经元，进而影响之后的输出。其中后一个结构是必要的，如果神经网络的后续输出与当前输出有着强相关性，且要使该神经网络可以递增地运行下去。然而有些反馈通道可能不会实现，依情况而定。

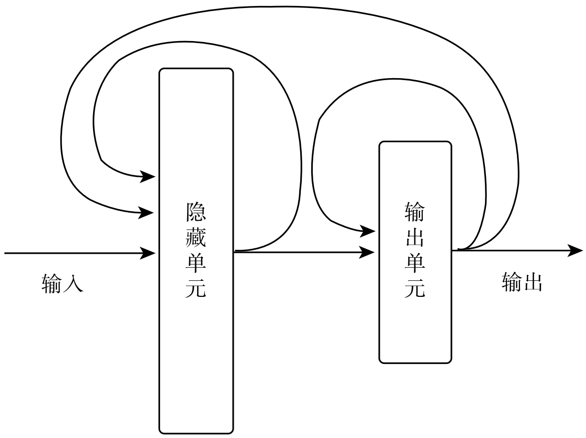


图 14-1　递归神经网络基本图解

现在通过一个实际的例子来理解递归神经网络。该递归神经网络没有输入层，有 4 个激励函数为 S 型函数的隐藏神经元和 2 个激励函数为线性函数的输出神经元。它沿着一个环形来训练参数（中心在原点，半径为 1，每 16 步转一圈，前 4 步有一个瞬变现象）。如图 14-2 所

示，在训练了几步后，这个系统大致沿着一个环形来运行，其 4 个隐藏神经元所对应的值如图 14-3 所示。

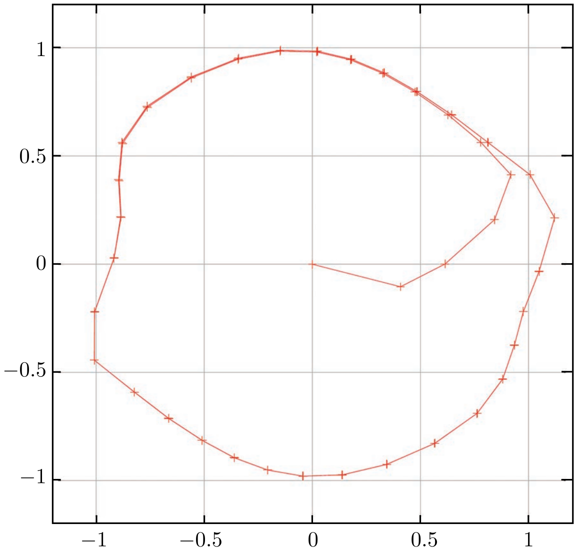


图 14-2 递归神经网络沿着环形训练：没有初始输入的情况下的输出序列

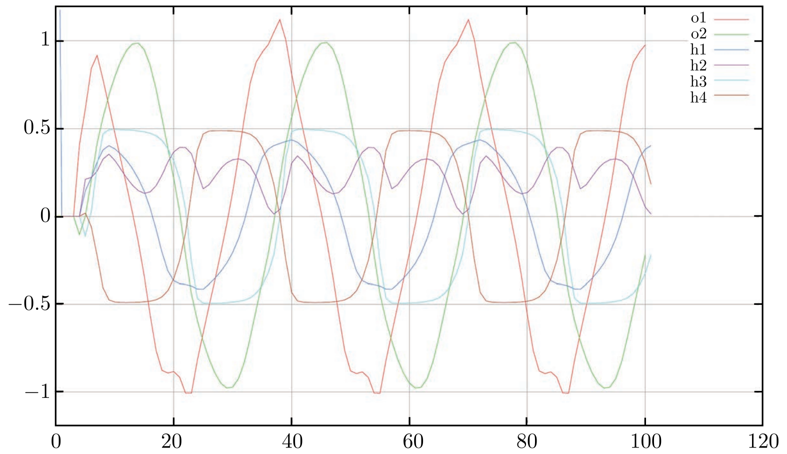


图 14-3 递归神经网络沿着环形训练：输出和隐藏层神经元（另见彩插）

递归神经网络里的环状结构有着重要的作用（最近的一篇综述见参考文献 [79]）。

- 递归神经网络能够沿着其循环连接的路径，动态地形成自维持的时序激活信号，即使没有任何输入也可以。递归神经网络就是一个**动力系统**，它的前馈网络就是其演变函数。

- 如果存在输入信号，那么递归神经网络将历史输入通过非线性转换后保存在内部状态中。递归神经网络具有动态的记忆功能，可以处理时序的内容。

从动力系统的角度来看，递归神经网络可以分成两类。第一类递归神经网络的特点是能量极小的动态随机和**对称连接**（神经网络的输出轨迹在某一合适的“能量”函数下达到其局部极小值，可看作梯度下降的一个变体），已知的实例有从统计物理学中衍生出来的**霍普菲尔德网络**（Hopfield network）[61]、玻尔兹曼机 [2] 和深度信念网络 [57]。这些学习系统的训练大多是**无监督的**，典型应用于联想储存器（其检索的内容对应能量函数的局部极小值）、数据压缩、数据分布的无监督建模和静态模式分类领域。对于每个输入实例，该系统会运行多次，最终达到某种收敛或平衡的状态。

第二类递归神经网络的特点是确定型动态更新和**有向连接**。应用该类网络可实现非线性的过滤器——将输入的时间序列转化成另一种时间序列。该类网络背后的数学原理是非线性动力系统，且该网络的训练是**监督的**。

14.2 能量极小化霍普菲尔德网络

霍普菲尔德网络（见图 14-4）——由参考文献 [61] 定义——是由一系列二值阈值神经元（即根据输入是否超过阈值，输出 1 或者 –1）构成的网络。网络中的神经元通过具有对称权重 w_{ij} 的边连接（没有自连接的边 $w_{ii}=0$）。神经元通过下述方式更新状态：

$$s_i \leftarrow \begin{cases} +1 & \sum_j w_{ij} s_j \geqslant \theta_i \\ -1 & \text{其他情况} \end{cases} \tag{14.1}$$

其中，s_i 是神经元 i 的输出状态，θ_i 是其阈值。神经元的状态更新分**异步**（随机选取一个神经元并更新）和**同步**（存在一个中心时钟，所有的神经元在同一时刻更新）两种方式。异步更新方式更符合生物学和物理学中的现象（物理学中的*自旋玻璃态*就是一种相关的模型）。初始的输入随着更新不断地改变，因此神经元的状态不仅仅依赖于初始状态，还依赖于之前一系列的更新过程。如果神经元的输出状态用 LED 灯表示（输出 +1 发光，输出 –1 不发光）——实际上本书的作者之一已经实现了这样的硬件——我们可以观察到 LED 灯随着时间闪烁的图案。现在主要的问题是：这样闪烁的图案可能有着怎样的意义？哪些计算可以在其上执行？对于霍普菲尔德网络而言，这些问题的答案与**优化一个合适的能量函数**有关 [在数学和物理学中也称作*李雅普诺夫函数*（Lyapunov function）]：

$$E = -\frac{1}{2}\sum_{i,j} w_{ij} s_i s_j + \sum_i \theta_i s_i$$

显然，当神经元被选中更新时，由于连接的权重是对称的，**能量函数E的值总是减小或不变**。多次重复更新后，霍普菲尔德网络最终会收敛于使得能量函数达到局部极小值的状态。能量

函数的局部极小值被称作霍普菲尔德网络的**稳定状态**。因此，LED 灯的闪烁图案最终会稳定下来并显示一个不变的图案。霍普菲尔德网络也可以“意味着”一个动力系统，它始于一个初始输入值，目标是**在初始输入值的吸引域中寻找一个局部极小值**，过程类似于一滴水通过流域盆地流入湖泊里。实际上，将输出值限制在一定范围内是非常必要的，否则求能量函数的极小值需要最小化二次型，而二次型的最小值有可能是无限的（负无穷）。

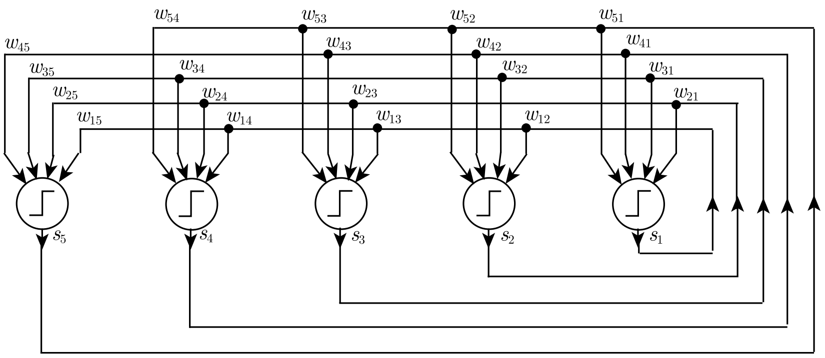


图 14-4 由 5 个神经元和反馈循环组成的霍普菲尔德网络

编程实现霍普菲尔德网络实际上就是**刻画其能级相图**（energy landscape）**中合适的局部极小值**。Donald Hebb 在 1949 年为解释“联想学习”而提出了 Hebbian 学习算法，它在训练霍普菲尔德网络时修改边的权重。在生物学中，神经元的同时激发现象会导致这些神经元之间的突触强度明显增加：“神经元同时发出信息，同时传递信息。如果神经元不能同步发出信息，那么一定是它们之间没有链接。” Hebbian 算法是局部的和递增的。对于霍普菲尔德网络来说，如果要学习 N 个二值模式，该算法可以通过如下方法实现：

$$w_{ij} = \frac{1}{N}\sum_{\mu=1}^{N} x_i^\mu x_j^\mu, \quad i,j = 1,\cdots,n$$

其中，每个模式 $\boldsymbol{x}^\mu = (x_1^\mu,\cdots,x_n^\mu)$ 都是一个长 n 位的序列，n 同时也是霍普菲尔德网络中的神经元数。在一个模式 $\boldsymbol{x}^\mu$ 中如果对应神经元 i 和 j 的位数是相等的，那么 $x_i^\mu x_j^\mu$ 的乘积将为正，即会使权重 w_{ij} 增加，所以神经元 i 和 j 的值一定是趋向相等的，若神经元 i 和 j 的值不等，则反之。

依霍普菲尔德网络中神经元的数量而定，Hebbian 算法能够在其能级相图中刻画出一系列局部极小值，前提是 N 不是很大（一个经验法则是，模式的数量 N 不超过神经元的数量 n 的 13.8%[55]），这些极小值可以很接近训练的 N 个模式。当更新规则 [式 (14.1)] 基于一个给定的模式 $\boldsymbol{x}^\nu$ 反复迭代时，该霍普菲尔德网络将会稳定在一个局部极小值上，因此，检索到的存储起来的模式与 $\boldsymbol{x}^\nu$ 最接近，如图 14-5 所示。

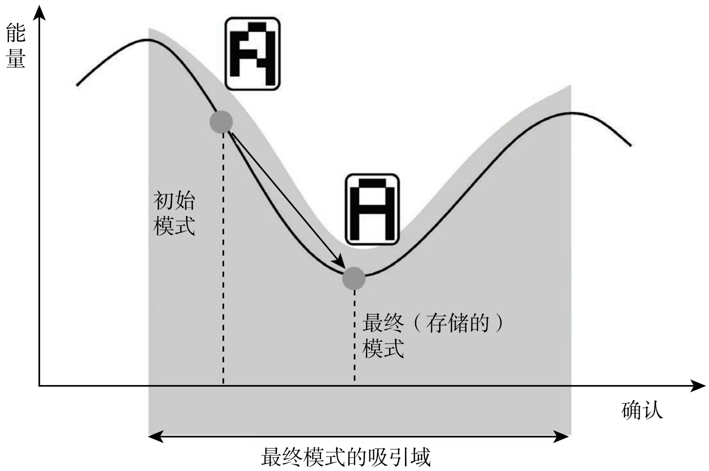


图 14-5　霍普菲尔德网络的能级相图，标出了其初始状态（曲线的上端），最终收敛的吸引状态和一个吸引域（阴影部分）

基于霍普菲尔德网络可以建立**内容可寻址存储**系统：只需提供模式内容的一部分（其他位随机设置），该网络就可以收敛于一个“记忆存储的”状态。这样的网络可以用来恢复失真的输入：在已训练的状态中找出与该输入最相似的状态。这也被称作**联想存储器**，类似于纠错码。霍普菲尔德网络连续权重的泛化方法与梯度下降类似，不过是沿着梯度的方向下降（但能量函数的值仍在下降）。

尽管霍普菲尔德网络具有重大的理论价值和研究意义，但在现实世界的应用中仍面临一些挑战，比如伪模式（spurious pattern）——收敛的局部极小值不对应训练的存储值——以及网络的容量限制。当霍普菲尔德网络存储了很多模式后，已存储的单元有可能与其他邻近的检索相混淆。显然，人类的记忆也具有相似的特性，语义上有关联的词语容易弄混，从而导致记忆存在误差。

14.3　递归神经网络和时序反向传播

现在来考虑更一般的递归神经网络——不要求对称的权重和二值输出（见图 14-1）。

这类递归神经网络可以展开为如图 14-6 所示的前馈网络，与标准的前馈神经网络的唯一区别在于，它允许输入神经元跳过隐藏层，直接连接输出神经元。训练这类递归神经网络的标准算法是**“时序反向传播”**（backpropagation through time, BPTT），主要思想是将前馈神经网络标准的反向传播应用于上述的展开模型 [113]。如果递归神经网络中一些神经元的输出可以反馈给其他神经元，那么由于**梯度消失和爆炸问题**[23]，它基于导数的训练方式会变得十分困难。梯度爆炸（exploding gradient）是指在训练递归神经网络时，由于模式的长时间依

赖，导致梯度远超其正常值，甚至能达到短时间依赖时的几何指数倍。梯度消失（vanishing gradient）恰恰相反，当长时间依赖的模式梯度呈几何指数式趋近 0 时，学习长时间相关的事件就变得不可能了。这个问题是由递归神经网络中神经元的迭代次数过多导致的，因为这会导致网络中一个很小的权重呈几何指数式增加或减小。参考文献 [85] 利用修改的时序反向传播算法，解决了上述问题（梯度基准剪切方法解决梯度爆炸，软约束解决梯度消失）。

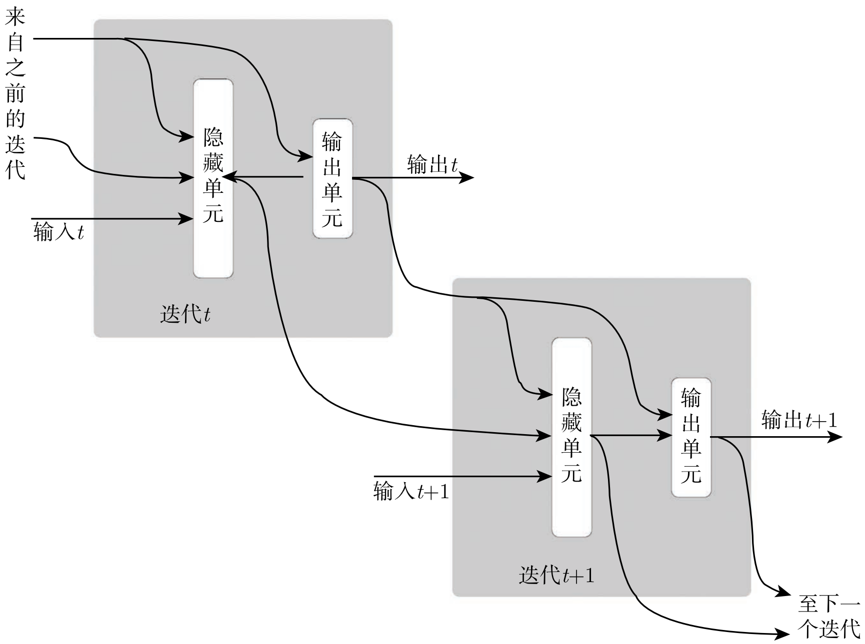


图 14-6　图 14-1 中的递归神经网络可以被瀑布状地展开成一个前馈网络，它的每一个前馈都是由当前迭代的输入和先前迭代的输入–输出组成

某种程度上，递归神经网络凸显了基于导数的优化方法的缺陷，推进了无导数方法的发展和训练体系的彻底改变，例如下面将要介绍的储备池学习方法和极限学习机方法。

14.4　递归神经网络储备池学习

递归神经网络（我们讨论的第二类，即没有对称权重的限制）是一个非常有前景的针对非线性时序应用的工具 [79]，它用一些容易接受和普遍的假设将生物学理论（大脑中存在递归连接的神经元）与动力系统相统一。

以往提出的很多训练方法都存在下述缺点。

- 在学习时不断改变神经网络的参数会导致网络动态**分岔**（bifurcation）：梯度信息退化，可能变得模糊不清，从而不能保证最终的收敛。

- 许多花销巨大的循环更新或许是有必要的，但会导致大型网络（超过 10 个神经元）的训练时间过长。
- 长期记忆的学习是困难的，因为所需计算的梯度信息随着时间呈几何指数式增加。
- 高级训练算法中的全局控制参数十分复杂，调参需要大量的技巧和先验知识。

21 世纪初，一种全新的方法分别以流体状态机[80] 和回声状态网络[67] 的面貌独立提出，这里统一将其称作**储备池计算**。储备池计算通过以下规则（见图 14-7）可以克服递归神经网络梯度下降算法的缺陷。

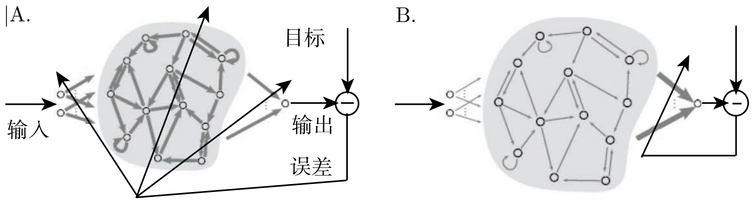


图 14-7 （左图）递归神经网络传统的基于导数梯度的训练方法需调整所有的连接参数（粗箭头），而（右图）储备池计算只需修改连接递归神经网络和输出之间的连接参数即可（改编自参考文献 [79]）

- 如果一个**随机生成的**递归神经网络在训练时不改变其参数，那么这个递归神经网络就称作**储备池**，它总是维持自身的状态不变，只被动地接受输入信号的刺激，将历史输入非线性地转化成输出。
- 最终期望得到的输出信号，由被刺激的储备池中所有的神经元输出信号**线性组合**而成，这里的线性组合可以通过线性拟合得到，例如最小二乘法。

储备池计算很快就成为递归神经网络建模的基本工具之一，它表现出更高的建模准确率，且对于连续时间和连续值的实时系统具有通用的建模容量。储备池计算也可以解释“为什么尽管存在有噪声的物理部分，但大脑仍可以进行精确的计算”。最后，一些递归神经网络可以通过向同一个储备池里添加更多的输出神经元来扩充，需干涉之前模型的设计功能。

在一些实例中，一个完全随机的储备池无法满足需求，因此现在的研究正向合适的储备池的生成和适应方法发展。储备池计算从蛮力随机方法出发，继而成为了一些不同方法的范例：(i) 生成或调整储备池；(ii) 训练储备池不同类型的读出。参考文献 [79] 是最新的一篇关于储备池计算综述。

14.5 超限学习机

还有一系列与储备池计算相关的关于前馈系统的研究，比如在多层感知机中，随机初始化各层参数，并只对最后一层进行线性回归训练。参考文献 [64] 中提出了**超限学习机**的概念，利用标准广义逆的方法计算最小二乘拟合（在 4.1 节中）。超限学习机和储备池计算都采用简

单却有效的线性回归算法，**仅通过修改输出权重**，就能解决传统神经网络训练算法中的许多问题，例如收敛到局部极小值、梯度消失或爆炸。它们主要的区别是，储备池计算的结构包含递归连接，从而实现短时记忆，而超限学习机使用纯粹的前向结构，也没有短时记忆。

实际上，在神经网络研究的萌芽阶段，人们就发现了类似于超限学习机的方法，即使随机初始化神经网络各层参数，也能得到不错的学习结果。例如，Rosenblatt [90] 和其他研究者偏爱随机选择输入特征探测器。E. Baum [20] 主张在模拟神经网络时，固定那些同一层中的连接参数，只调整那些连接不同层的参数。直到最近，参考文献 [65] 才提出了上述方法一般的理论和实践研究，并将其命名为**"超限学习机"**（Extreme Learning Machine，ELM）。

在线性代数中，众所周知，如果一个单层前馈网络（SLFN）具有 N 个隐藏神经元，并随机生成输入层参数和隐藏层偏差，那么由矩阵的逆可知，它恰好可以学习 N 个不同的观测样本，也就是说这些样本的矩阵是满秩的（线性无关的）。当然，该方法不能保证泛化的正确性，但可以使得单层前馈网络的学习只需要矩阵求逆这一简单且一次性的操作。

参考文献 [65] 中所做的工作严格证明了一点：如果隐藏层的激活函数是无限可微的，那么单层前馈网络的输入层参数和隐藏层偏差可以随机指定。当这些值被随机选定后，单层前馈网络可以被看作一个线性系统，其输出层参数可以通过对隐藏层输入矩阵进行简单的广义逆操作得到。目前已经有了各式各样的基于不同种类隐藏层神经元和隐藏层结构的泛化方法（见图 14-8）。某些情况下，超限学习机比传统的学习算法（比如反向传播算法）更快，得到的泛化结果也更好，尤其在网络中权重的范数被通常的二次罚分限制的情况下。为了得到最优的性能，超限学习机隐藏层的神经元数可以比反向传播算法多得多，这表明为了挖掘出足够多有效的神经元以确定输出，需要生产大量的随机神经元。参考文献 [63] 是最近的一篇相关综述。

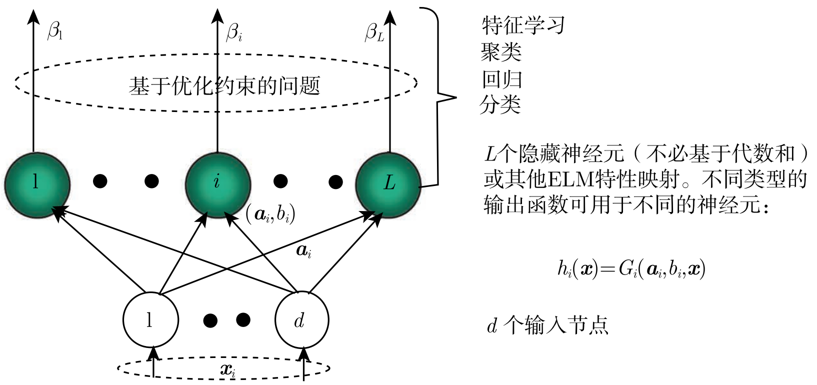


图 14-8 超限学习机的隐藏层可以包含不同种类的可计算节点（改编自参考文献 [63]）

参考文献 [68] 中的相关研究评估了物体识别的多阶段结构，并考虑了随机过滤器。参考文献 [114] 中提出了非传播（no-prop）的神经网络算法（类似于超限学习机，但采用迭代的最小二乘技术）。参考文献 [94] 中研究了随机参数的结构，并发现了一个确定的卷积池化结构（在图像处理中，空间上不同位置的神经元共享连接的参数）具有固定的选择频率和不变的转化效果，即使它的参数是随机的。基于这个发现，他们提出了**使用随机参数评估候选网络结构**，从而避免学习过程中过于耗时的方法。最新的神经网络算法中，网络的结构对提升算法性能的作用令人惊讶，尽管接下来的参数微调也的确能提升最终的算法性能。

显然，储备池计算和超限学习机都是沿着同一个研究方向进行的，虽然原始的储备池计算大多关注递归神经网络，而超限学习机关注前馈系统。参考文献 [32] 同时研究了储备池计算和超限学习机。



梗　　概

带有反馈回路的**递归神经网络**，可以使得“数学函数”（前馈网络）过渡到随时间进化并带有内部存储器的全面**动力系统**。

递归神经网络的机器学习是很难的，尤其是基于导数的方法。它所涉及的循环很多，可能会导致梯度爆炸或者消失。

最近提出的**储备池计算**（RC）和**超限学习机**（ELM）都采用一种激进的方法：与深度学习相反，它们生成大量的**随机构造块**（随机特征），并将模型的学习限制在一个最终的线性组合层中。具体来说，就是从储备池中挖掘有用的构造块，并将其适当组合起来，得到最终的学习结果。

鉴于生物神经元中的噪声影响，深度学习的导数方法难以实现，“随机构建辅以最终调参”的蛮力法的成功给予了我们解释大脑部件如何运行的希望，并使我们能够设计出更快且更灵活的机器学习算法。

我们很高兴生活在这样一个研究成果迸发的时期，各式各样疯狂的想法通过令人惊奇的情节转折和范式变化，推进机器学习和神经网络的前沿发展。

第二部分

无监督学习和聚类

第 15 章　自顶向下的聚类：K 均值

起初，神创造天地。地是空虚混沌，渊面黑暗，神的灵运行在水上面。神说，“要有光”，于是就有了光。神看光是好的，就把光暗分开了。神称光为昼，称暗为夜。······

神把用土所造成的野地各样走兽和空中各样飞鸟都带到那人面前，看他叫什么。那人怎样叫各样的活物，那就是它的名字。那人便给一切牲畜和空中飞鸟、野地走兽都起了名。

——《创世纪》



本章将开启一个新部分，也会进入一个新领域。到现在为止，我们考虑了监督学习方法，而这一部分的问题是：**在没有老师和标记的情况下，我们可以学到什么？**

像上面米开朗基罗的那幅画中发出的能量那样，我们正进入一个更具创造性的领域，其中包含关于探索、发现、意想不到的结果等概念。现在的任务不是亦步亦趋地跟着老师，而是自由地生成模型。很多情况下，自由不一定是人们所希望的，但它是继续前进的唯一途径。

想象一个孩子坐在电视机前。即使没有老师，他也会马上意识到好的电视机屏幕与坏了的屏幕之间的区别，因为坏了的屏幕会出现“雪花”般的随机噪声模式，而不是卡通片或国际新闻这些电视节目。更有可能的是，卡通片会使他更加兴奋，而不是国际新闻或随机噪声。正在工作的电视屏幕的画面（和世界的表面现象）并不是随机的，而是高度结构化的，根据某种显式或隐式的计划来安排。关于无监督学习的另一个例子，假设实体代表说不同语言的人，坐标与他们的口语音频测量值（如频率、振幅等）相关。在一个国际机场里，根据不同语言的

发声特点，大多数人可以很容易地识别出说不同语言的人群。例如，我们可以很容易地区分说意大利语和说英语的人，即使并不知道具体是哪种语言。

对结构（形式、模式、有趣事件的集合）的建模和理解，是我们认知能力的基础。 *名称*和语言的使用深深植根于大脑的组织能力。从本质上讲，名称是将不同经验组合起来的方式，使我们能够说话和推理。苏格拉底是人，*人皆有一死*，因此苏格拉底也会死[①]。

举例来说，被引入到**共同特征**推理的是动物的种类（以及相应的名称），而不是个别动物（“那人便给一切牲畜都起了名”）。在地理学中，大洲、国家、区域、城市、社区代表不同标度的地理实体的类别。聚类是非常人性化的活动，**将相似的东西放在一起，进行抽象**并且为对象类命名（见图 15-1）。想想把世界上的男人和女人分类，尽管个体之间差异显著，我们却对此相当有把握。

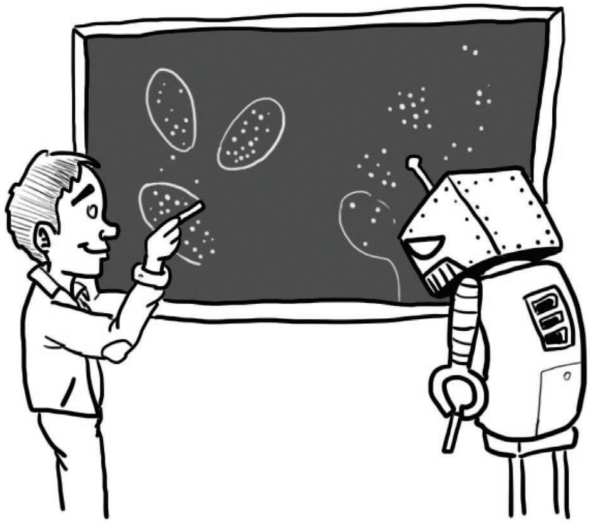


图 15-1　聚类深深地根植于分组和为对象命名的人类活动中

① 说实话，命名这种高度简化的方式让世界失去了神秘感，这就是用技术征服世界所付出的代价。Rainer Maria Rilke 在他的诗《我的庆典》（1909）中表达了这种观点。

我是如此害怕人类的语言，
它能如此准确地描述万物：
它称之为狗，它称之为房子，
在这开始，在那结束，
……
我想警告它：让任何事物如它们原本的样子！
我享受聆听它们发出的声音。
但你总是干预它们：它们变得安静和静止。
这就是你怎么扼杀它们的。

聚类必须处理**信息的压缩**。当数据量超过人们的消化能力时，就会发生**认知过载**，我们大脑中有限的“工作记忆”也不足以应付这项工作。实际上，为了减少用于分析的数据点数量，可以使用过滤器来限制数据值范围。但这并不一定是最好的选择，因为这种情况下是在各个坐标上筛选数据，然而**更加全局的图景**或许才是人们希望看到的。

聚类方法以智能和数据驱动的方式收集类似的点并放在一起，因此人们的注意力可以集中于一个相对小但相关的**原型**集合。原型概括了自身所代表的实例子集中的信息。当类似的实例被组合在一起时，人们可以对这些分组而非个别实体进行推理，因此减少了不同的可能性的数目。

可以想象，聚类的实际应用是没有止境的。举一些例子，在**市场细分**中，人们将广阔的目标市场划分为有共同需求的客户子集，然后执行针对每一类客户的共同需求和期望的策略。在**金融领域**，聚类将有类似行为的一批股票分在一组，可以提升投资的多样性并降低风险。在**医疗保健**中，可以将疾病按其症状进行聚类。在**文本挖掘**中，根据所分析的文本的结构和含义，将单词分组。**语义网络**可以表示概念之间的语义关系。它是一个有向或者无向图，包含代表概念的顶点和边。由于存在不同的关系（例如“是”“有”“住在”，等等），也就不存在统一的方法来对实体进行分组。

15.1　无监督学习的方法

鉴于聚类方式的创造力以及分类对象的不同，聚类的方法也千差万别。传统上将这些方法分为自顶向下和自底向上两种。

在**自顶向下或分裂聚类**中，首先确定都有哪些类，再将不同的实例分类，目标是把类似的实例分在一起。注意，这些类别没有标签，只存在如何细分的问题。试想在固定数量的抽屉柜子中如何摆放衣服。如果你是一个成年人（如果你是一个欢乐的少年，请询问你的父母），你最后可能会把袜子与袜子放在一起，衬衫与衬衫放在一起。

在**自底向上**或**凝聚聚类**中，数据分类是自然形成的，人们可以直接开始合并（关联）最相似的条目。一旦创建了较大的条目分组，我们就合并最相似的组，以此类推。若分组是有意义的，就停止该过程，当然这取决于具体的度量、应用领域和用户的判定。最终的结果是越来越大的集合构成的层次组织（称为树状图），体现了逐渐变大的合并。树状图在自然科学中是常见的结构，想想动物学或植物学物种的组织层级关系。

有一种更先进和灵活的无监督策略，称为**维数降低**：为了减少描述一组试验数据的坐标数，需要理解结构和不同实例“变化的方向”。如果对人的面部聚类，变化的方向可以和眼睛颜色、人中长短、鼻子和眼睛之间的距离等参数相关。所有类型的面部可通过改变几十种参数来获得，而这肯定比一张图像中的像素总数要少得多。

另一种为一组实例建模的方法是，假设它们是由一个相关的**概率过程**产生的，那么对过程建模就是理解结构和不同类别的一种方式。**生成模型**的目的是确定产生观察实例的过程的

概率分布并建模。想想通过对不同作者使用的主题和字词建模来对图书进行分组（事先不知道作者姓名）。一个作者有一定的概率会选择某个主题。主题确定之后，与主题相关的字词将以特定的概率产生。当然，这样是不会产生杰作的，但会产生类似的字词出现的最终概率，很多情况下足以识别出一个未知的作者。

我们的视觉系统对图像的显著部分进行聚类的功能是极其强大的。**可视化**，例如线性或非线性**投影**到低维空间（通常具有两个维度），可能对“手动”——好吧，其实是“眼动”——确认结构和聚类是很有效的。

最后，兼具趣味性和挑战性的应用需要将有监督和无监督策略（**半监督学习**）相结合。想想一个“大数据”的应用：将不计其数的网页聚类。标记可能非常昂贵（因为可能需要人工为页面分类），因此非常少见。将一组大量的未标记网页加入稀疏标记的网页集合中，可大大提高最终结果的准确率。

对这部分内容做了整体介绍之后，本章将侧重于一种应用广泛且有效的自顶向下的方法，称为 **K 均值聚类**。

15.2 聚类：表示与度量

聚类中存在两种不同的情景，这取决于要进行聚类的实体是如何组织的（见图 15-2）。某些情况下可根据每个实体的**内部表示**（对于实体 d，通常是一个 M 维向量 $\boldsymbol{x}_d$）推导出实体间的相异性或相似性。这种情况下可以为每一个类推导出**原型**（**或中心**），例如通过计算所包含的实体（向量）特性的平均值。其他情况下，可用的只有一个相异性的**外部表示**，所得模型是由边连接实体的无向加权图。

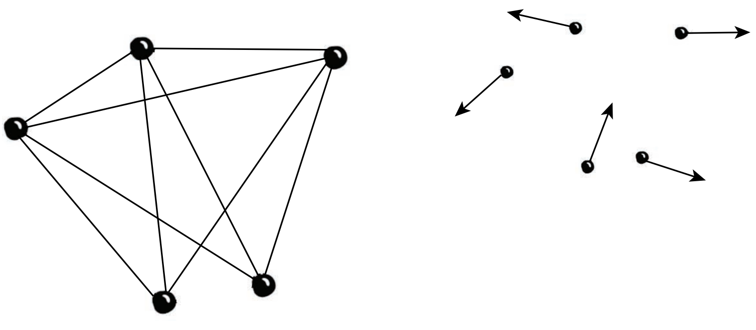


图 15-2 基于关系的外部表示（左图）和基于坐标的内部表示（右图），分别表示一对数据点间的相似度和每个独立数据点对应的向量

例如，假设有市场调查表明，超市会在邻近的货架上摆放相似的食物，因为这样可以带来更多的盈利。特定食品的内部表示可以用数字向量来描述：食物类型（1= 肉，2= 鱼 ⋯⋯）、

卡路里含量、颜色、包装大小，以及建议食用日期等。相似性则可以通过欧氏度量或比较对应向量的标量积来得出。

通过询问顾客可以得到外部表示，让他们对商品 X 和 Y 的相似性进行评级（要有固定的分值范围，例如 0~10），然后对顾客的投票求平均，导出外部相似性。

聚类方法的有效性依赖于**相似性度量**（如何衡量相似性），而相似性度量与需要解决的问题相关。传统的欧几里得度量在某些情况下适用，这需要不同的坐标上的测量单位相近，且有一个可参考的显著性水平；如果使用不同的测量单位，欧几里得度量就不适用了。例如，如果一名警察在识别面部时，以毫米为单位来测量眼距，而以千米为单位来测量人中距离，那么欧氏度量几乎毫无意义。同样，如果房地产市场有房屋的颜色数据，在根据商业目的对房屋进行聚类时，颜色不会很重要。但是，将不同艺术家画的房屋进行聚类时，颜色就变得非常重要。度量确实是针对特定问题的，这就是我们把实体 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 之间的差异写成 $\delta(\boldsymbol{x},\boldsymbol{y})$ 的原因，留待以后确定如何具体计算它的。

如果存在一个内部表示，可以由通常的**欧几里得距离**导出度量：

$$\delta_E(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{x}-\boldsymbol{y}\| = \sqrt{\sum_{i=1}^{M}(x_i-y_i)^2} \tag{15.1}$$

在三维空间中，这是传统的距离，通过先将边取平方然后求平方根来测量。符号 $\|\boldsymbol{x}\|_2$ 表示向量 $\boldsymbol{x}$ 的欧几里得范数，而下标 2 通常被删去。

另一个值得注意的度量是曼哈顿距离或出租车范数，之所以这么称呼，是因为它测量一辆出租车在一个长方形的街道网格中从原点到 x 点的距离：

$$\delta_{\text{曼哈顿}}(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{x}-\boldsymbol{y}\|_1 = \sum_{i=1}^{M}|x_i-y_i| \tag{15.2}$$

像往常一样，没有绝对正确的范数，也没有绝对错误的范数：对于每个问题，范数必须适当地反映距离度量。在纽约的出租车喜欢曼哈顿范数，飞行员则更喜欢欧几里得范数（至少短距离是这样，而由于地球的曲率，仍然需要基于测地学的不同距离度量）。在某些情况下，只有评价了聚类结果*之后*才能认识到什么是测量距离的适当方法，这也是工作中创造性和开放性的来源。

还有一种可能性是，从两个归一化向量的标量积给出的相似性出发，然后取逆以获得相异性。具体来说，向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 之间的归一化点积，类比二维和三维的几何，可以理解为它们之间角度的余弦值，因此被称为**余弦相似性**：

$$\text{余弦相似性}(\boldsymbol{x},\boldsymbol{y}) = \cos(\theta) = \frac{\boldsymbol{x}\cdot\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|} = \frac{\sum_{i=1}^{M} x_i\ y_i}{\sqrt{\sum_{i=1}^{M}(x_i)^2}\ \sqrt{\sum_{i=1}^{M}(y_i)^2}} \tag{15.3}$$

然后取逆得到相异性：

$$\delta(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x}\|\|\boldsymbol{y}\|/(\epsilon + \boldsymbol{x} \cdot \boldsymbol{y})$$

其中 ϵ 是为了避免零作为除数而加入的一个小量。

注意，余弦相似性仅取决于两个向量的**方向**，如果每个坐标值都乘以一个固定的数，余弦相似性不会发生变化。而如果一个向量乘以一个标量值，那么欧几里得距离会发生变化。标准欧几里得距离的缺点是，不同的坐标值可以有非常不同的取值范围，导致距离可能被坐标的某一个子集所支配。这可能发生在以不同的方式选取测量单位的情况下，例如有的坐标以毫米为单位，有的以千克为单位，还有的以千米为单位：如果分析的关键取决于挑选一套合适的物理单位，那么它始终会令人非常不愉快的。为了避免这种麻烦，我们需要没有物理单位的**无量纲**（dimensionless）值。此外，不妨对测量值进行归一化，让所有的值在测量距离之前都处于 0~1 的范围。

上述方法可以用下面的定义实现：

$$\delta_{\text{范数}}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{i=1}^{M} \left(\frac{x_i - y_i}{\text{maxval}_i - \text{minval}_i} \right)^2} \tag{15.4}$$

其中 M 是坐标数，minval_i 和 maxval_i 分别是所有实体的第 i 个坐标所能达到的最小值和最大值。

一般情况下，可以确定一个正定矩阵 $\boldsymbol{M}$ 来变换原始度量：

$$d_{ij} = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}} \boldsymbol{M} (\boldsymbol{x}_i - \boldsymbol{x}_j)}$$

马氏距离（Mahalanobis distance）是一个例子，它考虑数据组的相互关系，而且是标度不变的（即使用不同的单位进行测量，如毫米或千米，它也不会改变）。马氏距离将在后面的章节中详细讨论。

15.3 硬聚类或软聚类的K均值方法

硬聚类（hard clustering）是将实体集 D 分割为 k 个不相交的子集 $C = \{C_1, \cdots, C_k\}$，以达到下面的目标。

- 最小化平均聚类内部的相异性：

$$\min \sum_{d_1, d_2 \in C_i} \delta(\boldsymbol{x}_{d_1}, \boldsymbol{x}_{d_2}) \tag{15.5}$$

  如果存在一个内部表示形式，那么聚类的中心 $\boldsymbol{p}_i$ 可以通过对第 i 类所有成员的内部表示向量求平均值，来得到 $\boldsymbol{p}_i = (1/|C_i|) \sum_{d \in C_i} \boldsymbol{x}_d$。

这些情况下，聚类内部的距离可以用相对于聚类中心 $\mathbf{p}_i$ 的距离进行测定，从而得到相关但不同的最小化问题：

$$\min \sum_{d \in C_i} \delta(\boldsymbol{x}_d, \boldsymbol{p}_i) \tag{15.6}$$

- 聚类间距离的最大化。人们希望不同类别可以相互区分清楚。

正如所料，这两个目标并不总是互相兼容的，聚类确实是一个**多目标优化**任务。目标的重要性留给终端用户来权衡：究竟是要聚类内部的实体尽可能相似，还是要聚类之间的区别更明显，当然这也取决于选定的聚类的数目。

分裂算法（divistve algorithm）是最简单的聚类算法之一。这类算法从整个集合开始，陆续把它分成更小的聚类。一个简单的方法是一开始就决定聚类的数目 k，然后将数据细分为 k 个子集。如果效果不理想，就重新选择 k 值，然后再运用该算法。

如果想用单个向量来表示一组实体，合适的做法是选择使平均**量化误差**（quantization error）最小化的原型，这种误差是用原型取代实体时产生的：

$$\text{量化误差} = \sum_d \|\boldsymbol{x}_d - \boldsymbol{p}_{c(d)}\|^2 \tag{15.7}$$

其中 $c(d)$ 是 d 所在的类。

在统计和机器学习中，**K 均值聚类**将样本划分为 k 个由**中心**（类 c 的原型表示为 $\boldsymbol{p}_c$）所表示的类，每一个实例属于中心与该实例最接近的类。迭代方法被用来确定 K 均值中的原型，如图 15-3 所示，包含下列步骤。

(1) 选择聚类数 k。

(2) 随机生成 k 个聚类，并确定聚类中心 $\boldsymbol{p}_c$，或直接产生 k 个随机点作为聚类中心（换言之，从原始数据点随机选择初始中心位置）。

(3) 重复以下步骤，直到满足某个收敛标准，通常是最后一次分配没有改变，或已经达到迭代的最大次数。

　(a) 将每个点 $\boldsymbol{x}$ 分配到中心最近的聚类，即最小化 $\delta(\boldsymbol{x}, \boldsymbol{p}_c)$ 的那个。

　(b) 通过求上一步中分配的点的平均值来重新计算新聚类的中心：

$$\boldsymbol{p}_c \leftarrow \frac{\sum_{\text{聚类 } C \text{ 中的实体}} \boldsymbol{x}}{\text{聚类 } C \text{ 中的实体数}}$$

该算法的主要优点是简单快速，可以用在很大的数据集上。**K 均值聚类**可以看作**期望最大**（EM）算法的一个精简版本[①]：如果实例到聚类的分配是已知的，那么就可以算出中心；另

① 在统计学中，使用 EM 算法在含有隐变量的统计模型中寻找最大似然值或最大后验估计量。EM 算法是一个迭代方法，交替执行期望（expectation）步骤和最大化（maximization）步骤：在期望步骤中使用隐变量的当前估计值计算模型的对数似然期望，在最大化步骤中计算使得期望步骤中似然期望最大的参数值。

一方面，一旦中心是已知的，聚类分配就很容易计算了。因为一开始聚类中心（各聚类的参数）和成员分配都是未知的，所以通过分配和中心参数重新计算这两个步骤的循环来达到一致的状态。

给定一组原型，一个有趣的概念是**沃罗诺伊图**（Voronoi diagram）。每个原型 $\boldsymbol{p}_c$ 被分配到一个沃罗诺伊单元，单元中包含相比于其他原型，离 $\boldsymbol{p}_c$ 更近的所有点。沃罗诺伊图的分割线是空间中与两个最近的中心距离相等的所有点。沃罗诺伊节点是与 3 个（或多个）中心等距的点。图 15-3 给出了一个例子。

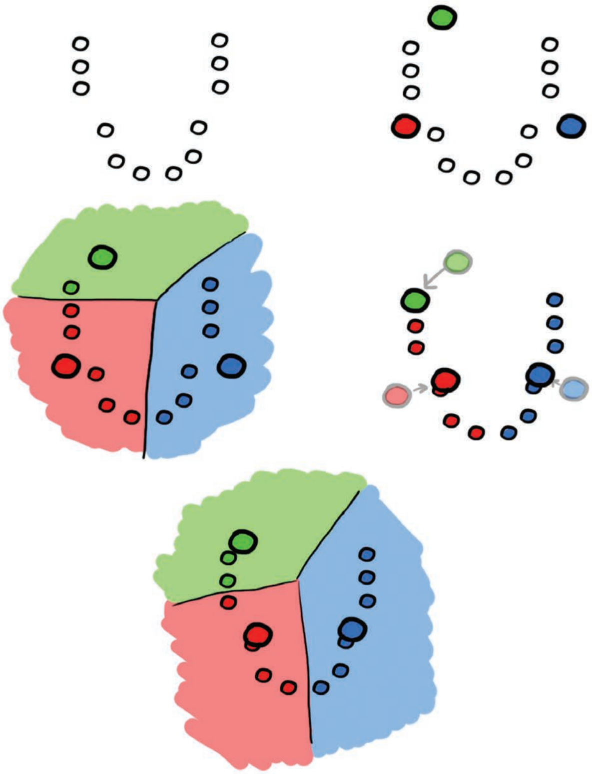


图 15-3 K 均值算法示例（从上到下，从左到右），初始的中心点如图所示，空间被细分成靠近中心点的各个部分（沃罗诺伊图：每一部分内的点都离给定的中心点最近），然后计算新的中心点，继续进行细分

目前为止，我们都在考虑硬聚类，也就是说实体都被刚性地分配。然而在某些情况下，较软的方法会比较合适，也就是说**分配不是刚性的，而是概率的或模糊的**。每个实体的分配是根据其被分为不同的类的概率（或模糊值）来定义的，因此这些值的和为 1。例如，考虑秃顶与非秃顶的人的聚类。如果把一位还有几根头发的中年男子归为秃顶，他或许会觉得自己没

有被温柔相待。顺便说一句，这种情况下也不宜谈论秃顶的概率，还是模糊隶属度比较适合：人们可能认为该人属于秃顶人群的模糊值为 0.4，而属于头发较多的人的模糊值为 0.6。

在**软聚类**（soft clustering）中，聚类成员可以被定义为相异性的递减函数，例如：

$$\text{成员资格}(\boldsymbol{x}, c) = \frac{\mathrm{e}^{-\delta(\boldsymbol{x},\boldsymbol{p}_c)}}{\sum_c \mathrm{e}^{-\delta(\boldsymbol{x},\boldsymbol{p}_c)}} \tag{15.8}$$

更新聚类中心的方法可以是**批次更新**或**在线更新**。在线更新时，人们反复考虑一个实体 $\boldsymbol{x}$，例如通过从整个集合中随机抽取，推导它的当前模糊聚类成员资格，更新所有的原型，使得原本接近的原型更接近给定的实体 $\boldsymbol{x}$：

$$\Delta\boldsymbol{p}_c = \eta \cdot \text{成员资格}(\boldsymbol{x}, c) \cdot (\boldsymbol{x} - \boldsymbol{p}_c) \tag{15.9}$$

$$\boldsymbol{p}_c \leftarrow \boldsymbol{p}_c + \Delta\boldsymbol{p}_c \tag{15.10}$$

用物理来类比，上述等式中的原型被每个实体牵着，沿向量 $(\boldsymbol{x} - \boldsymbol{p}_c)$ 移动，力的大小与成员资格成正比，因此会更为接近 $\boldsymbol{x}$。

在批量更新中，首先对所有实体对更新的贡献进行求和得到 $\Delta_{\text{total}}\boldsymbol{p}_c$，然后进行更新，如下所示：

$$\boldsymbol{p}_c \;\leftarrow\; \boldsymbol{p}_c + \Delta_{\text{total}}\boldsymbol{p}_c \tag{15.11}$$

如果参数 η 很小，两种更新的结果往往会非常相似；当 η 增加时，结果会产生差异。在线更新避免在移动原型之前对所有的贡献求和，因此当数据点数量变得非常大时，建议使用在线更新。

K 均值的结果可以用散点图来进行可视化，如图 15-4。k 个聚类原型都标有灰色大圆圈。某个类中的数据点是那些在所有 k 个原型中与给定的原型最接近的。

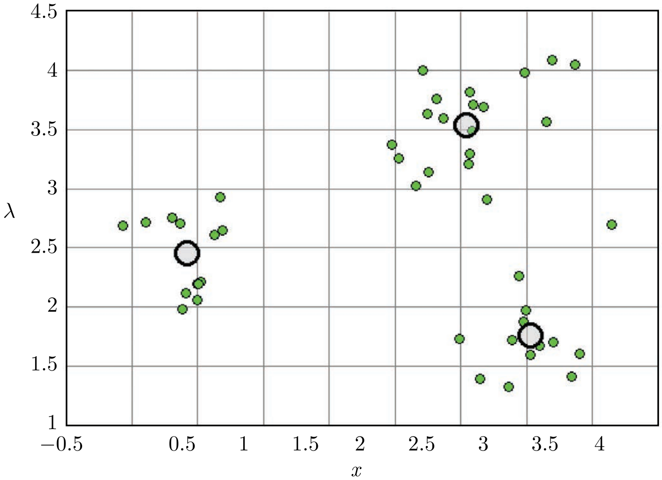


图 15-4　K 均值聚类，独立的数据点和聚类原型如图所示



梗　概

无监督学习仅用输入数据建立模型，不考虑分类标签。具体来说，聚类的目的是把相似的实例分在同一组，不同的实例分在不同的组。开始时聚类的信息可以由点之间的关系（**外部表示**）给定，或者由描述各个点的向量（**内部表示**）给定。第二种情况下，平均向量可以用作聚类成员的**原型**。

聚类的目标是：通过抽象化来压缩信息（考虑群体而不是个体成员），确定实验点（通常不是随机分布在输入空间，而是在某些区域“聚集”）的整体结构，并通过使用原型来降低认知超负荷。

不存在所谓“最好”的聚类准则。结果是否有趣，依赖于测量**相似性**的方式和用于后续步骤的分组的相关性。人们尤其需要对两个目标进行权衡：同一个类中的成员相似性高，不同类的成员的相异性高。

自顶向下的聚类中，首先选择所需要的类的数量，然后对实例进行细分。K 均值聚类一开始先设置 k 个原型，将实例分配到最近的原型，之后用分配的实例的平均值来重新计算原型

聚类提供了一个新的角度来看待你的狗，托比。狗是一类活的生物体，有 4 只爪子，会吠叫，开心的时候会摇尾巴。而托比是你最喜欢的小宠物的所有相关经验和情感的聚类。

第 16 章　自底向上（凝聚）聚类

羽毛相同的鸟聚集在一起。
（谚语：物以类聚，人以群分。）



一般情况下，聚类方法需要设置许多参数，如第 15 章中提到的 K 均值聚类，需要选择适当的聚类数目。避免一开始就选择聚类数目的一种方法是逐层构建更大的聚类，并把选择最合适的聚类数目和大小的任务留给接下来的分析阶段。这就是所谓**自底向上**或**凝聚聚类**。分层算法利用已经建立的聚类找到接下来的聚类，迭代开始时把每个元素作为一个聚类，然后将它们逐渐合并成更大的聚类。每一步中都会选择最为相似的聚类进行合并。

16.1　合并标准以及树状图

令 $\mathcal{C}$ 表示当前的聚类，它是实体集的子集——单个聚类 C——的集合。那么 $\mathcal{C}$ 可以定义一个划分：每个实体属于一个且只属于一个类。最初，$\mathcal{C}$ 中只有单元集，即每个集合只有一个实体。

正如自顶向下的聚类，自底向上的合并也需要距离的度量来指导聚类过程。这种情况下，相关的度量是两个类 $C, D \in \mathcal{C}$ 之间的距离，称之为 $\bar{\delta}(C, D)$，这个距离是从原来的实体之间

的距离 $\delta(x,y)$ 派生出来的。我们至少有 3 种不同的方式来定义它，不同的定义方式会导致不同的结果。实际上，考虑数据对之间距离的平均值、最大值或最小值都是可行的，如下所示：

$$\bar{\delta}_{\text{ave}}(C,D)=\frac{\sum_{x\in C,\ y\in D}\delta(x,y)}{|C|\cdot|D|}$$
$$\bar{\delta}_{\min}(C,D)=\min_{x\in C,\ y\in D}\delta(x,y)$$
$$\bar{\delta}_{\max}(C,D)=\max_{x\in C,\ y\in D}\delta(x,y)$$

算法现在继续执行下面的步骤：

(1) 在当前的 $\mathcal{C}$ 中查找距离 $\bar{\delta}^*=\min_{C\neq D}\bar{\delta}(C,D)$ 最短的 C 和 D；

(2) 用 $C\cup D$ 替代 C 和 D，并且将 $\bar{\delta}^*$ 记作该合并发生时的距离；

直到得到包含所有实体的单一聚类。

层次合并过程的历史和各种合并操作发生时的距离值可以用来绘制**树状图**（dendrogram，源自希腊语 dendron“树”和gramma“绘图”），以视觉的方式来展示合并过程，如图 16-1 和图 16-2 所示。

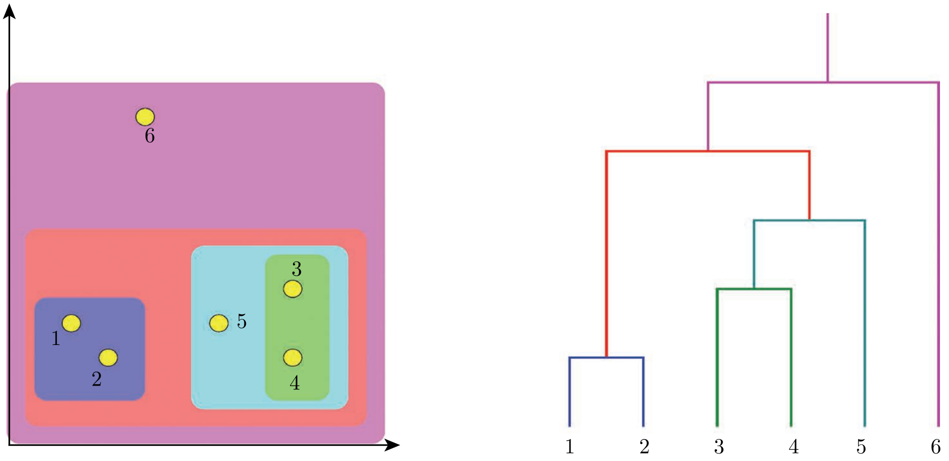


图 16-1 二维空间中数据点自底向上聚类示意图（使用标准欧几里得距离），每个数据点都由两个数值构成（另见彩插）

树状图是一个树状结构，其中底部是原始的实体，并且用水平线来连接两个融合的类来表示每一次的合并。水平线的纵轴坐标值显示了合并发生时对应的 $\bar{\delta}^*$ 值。为了重构聚类过程，想象一下你在这个树状图中从底部开始向上移动一把水平标尺。作为树状图的近亲，树在自然科学中被用来直观地表示相关的物种，其中根代表最古老的共同祖先，分支表示之后分裂产生不同的物种。

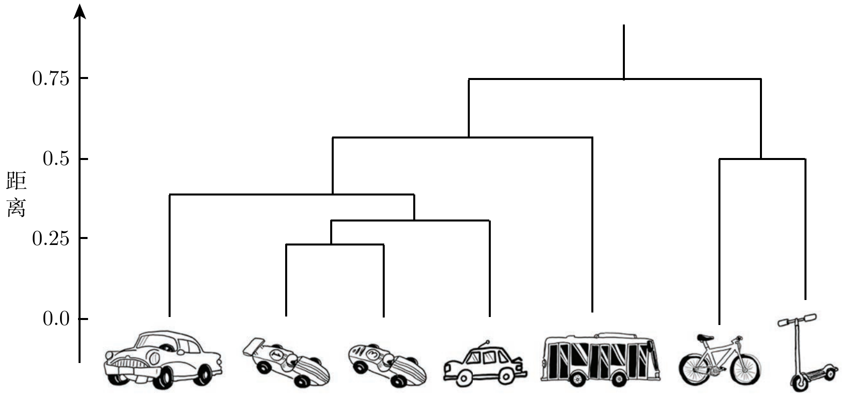


图 16-2　交通工具的自底向上聚类示意图

选择所期望的距离级别的值，并据此来水平切割整个树状图，马上就得到了该距离级别的聚类数目，这些子树中的叶子即是这些聚类中的成员。这种方式提供了一种简单的视觉机制，可以用于分析层次结构，并根据具体的应用和树结构来确定合适的聚类数目。例如，如果沿着树形图的纵轴能找到一个大的水平距离缺口，那么这是一个不错的水平切断级别，可以确定“自然”的聚类。

16.2　适应点的分布距离：马氏距离

基于测量确定头骨相似性的问题（1927 年）促使**马氏距离**得到发展，现在马氏距离广泛应用于**在测量相异性时加入数据分布的考虑**。数据分布由相关矩阵来建模。

将一些点归为一类之后，人们希望能（整体）定量地描述整个类，而不是简单地将其看作像云一样聚在一起的点。接下来，假定形成聚类的云状点集形式为简单的球形或椭球形，暂时不包括更复杂的形式，例如螺旋形、之字形或类似卷积的形式。

此外，在 N 维欧几里得空间中给定一个新的测试点，人们希望估计新的点属于该类的概率。第一步可以从寻找的样本点的平均值或中心开始。直观来说，所讨论的点越接近这个中心，越有可能属于该类。

然而，我们也需要知道这些数据分布的范围是大还是小，这样对于某个给定的点到中心的距离，我们才能决定它是大还是小。简单的做法是用抽样点到质心（center of mass）的距离来估计标准差 σ。如果测试点与质心之间的距离小于一个标准差，那么可以得出结论：新的测试点有很高的概率属于该类。这种直观的方法可以进行量化，定义测试点和样本集之间的**归一化距离** $(x-\mu)/\sigma$ 即可。将此代入正态分布，可以推导出该测试点属于该集合的概率。

上述方法的缺点是，它假定样本点以球形的方式分布。如果分布是高度非球形的，如椭球形，那么该测试点属于该类的概率不仅仅取决于到质心的距离，同时也取决于方向。在该

椭球形短轴的方向上，测试点必须近一些，而在长轴的方向上，测试点可以稍远一些。

最能代表该集合概率分布的椭球形可以通过构建样本的**协方差矩阵**进行估计。

令 $C=\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n\}$ 的是 D 维的一个类。类的中心 $\bar{\boldsymbol{p}}$ 是类平均值：

$$\bar{\boldsymbol{p}}=\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i \tag{16.1}$$

令协方差矩阵中的元素定义为：

$$S_{ij}=\frac{1}{n}\sum_{k=1}^{n}(p_{ki}-\bar{p}_i)(p_{kj}-\bar{p}_j),\quad i,j=1,\cdots,D \tag{16.2}$$

马氏距离是从椭球体的质心到测试点的距离简单地除以该椭球体在测试点方向上的宽度。图 16-3 说明了这一概念。具体来说，如果一组值的均值是 $\boldsymbol{\mu}$、协方差矩阵是 $\boldsymbol{S}$，向量 $\boldsymbol{x}$ 到这一组值的**马氏距离**定义为：

$$D_M(\boldsymbol{x})=\sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{S}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{16.3}$$

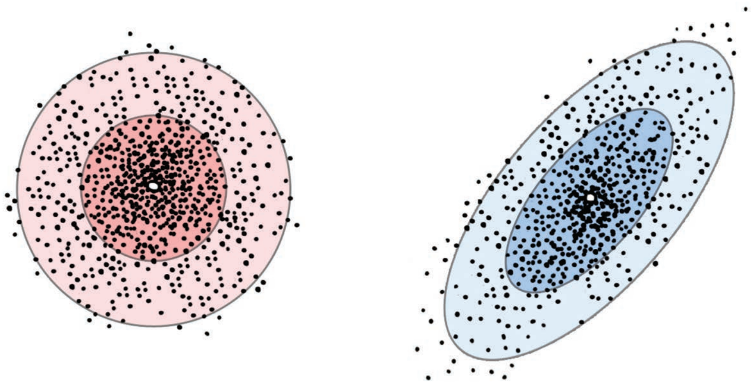


图 16-3 左图使用欧几里得距离作为差异性度量，而右图使用马氏距离，因其数据点分布呈椭球形

马氏距离也可以定义为两个随机向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 之间的差异性度量，这两个随机向量都服从协方差矩阵为 $\boldsymbol{S}$ 的同一个分布：

$$d(\boldsymbol{x},\boldsymbol{y})=\sqrt{(\boldsymbol{x}-\boldsymbol{y})^{\mathrm{T}}\boldsymbol{S}^{-1}(\boldsymbol{x}-\boldsymbol{y})} \tag{16.4}$$

如果协方差矩阵是单位矩阵，马氏距离就简化为欧几里得距离。如果协方差矩阵是对角矩阵，那么得到的距离度量被称为**归一化的欧几里得距离**：

$$d(\boldsymbol{x},\boldsymbol{y})=\sqrt{\sum_{i=1}^{D}\frac{(x_i-y_i)^2}{\sigma_i^2}} \tag{16.5}$$

其中 σ_i 是 x_i 在样本集上的标准差。

弄清了马氏距离的概念，以及通过由到质心的距离所确定的马氏距离度量的椭球体，我们可以描述聚为一类的云状点集，并理解聚类可视化的方法。

16.3　附录：聚类的可视化

本节介绍如何在三维空间中可视化聚类（跳过本节不会影响对后面章节的理解）。为了以图像的方式表示聚类，可以将它的**惯性椭球体**可视化，表面由到该聚类平均位置距离为单位距离的点的轨迹组成，距离度量是描述该类别的马氏度量。开始时将数据点投影到三维空间，并计算相应的 3×3 的协方差矩阵。

图形软件包的三维渲染中，点可以用 $\mathbb{R}^4$ 中的齐次坐标系的行向量来表示，无限平面表示为 $(x, y, z, 0)$，将单位球体映射成所需的椭球体的投影坐标变换由下面的矩阵表示：

$$T_C = \begin{pmatrix} S_{11} & S_{12} & S_{13} & 0 \\ S_{21} & S_{22} & S_{23} & 0 \\ S_{31} & S_{32} & S_{33} & 0 \\ \bar{p}_1 & \bar{p}_2 & \bar{p}_3 & 1 \end{pmatrix} \tag{16.6}$$

当上下移动层次聚类的级别时，类 C 将拆分成若干个类 $C_1, \cdots, C_l$。为了在脑海中形成正确的图像，可以将椭球体 T_C 与其 l 个后代 $T_{C_1}, \cdots, T_{C_l}$ 的参数化过渡想象成一组动画，并且椭球体

$$T_{C_i}^{\lambda} = (1-\lambda)T_C + \lambda T_{C_i}, \quad i = 1, \cdots, l$$

可以被画出来，其中参数 λ 在一个给定的时间段内（目前是 1 秒钟）均匀地从 0 变为 1。这将有效地展现出原始的椭球体变形（morphing）成它的后代的过程。

图 16-4 展示了分析一组汽车得到的聚类，其特征由含有机械特性和价格的向量给定。边的强度与物距有关：物距越近，颜色越深。

人们可以上下浏览聚类的层次结构，直到确认用于分析的合适的聚类数目。接下来就可以对原型进行检查，进而概括地描述这些数据。

在这种方式下，一种特别有用的浏览工具是**平行坐标**展示。一个将费希尔鸢尾花数据集分成 3 类的例子如图 16-5 所示。在平行坐标图中，每个垂直轴对应着数据的一维属性，一个 n 维空间的数据点可以表示成一条折线（由几条线段组成），它的每个端点都在平行的轴上，第 i 个端点在轴上的位置对应于数据中第 i 维度的值。使用过滤器可以只显示我们感兴趣的数据，而且可以调整轴的顺序、线和背景色等属性，让图片更加美观。一眼望去，无论是单个的点，还是整体的聚类结构（或项集的子集），视觉上都非常直观。

Ferrari 458 ltalia

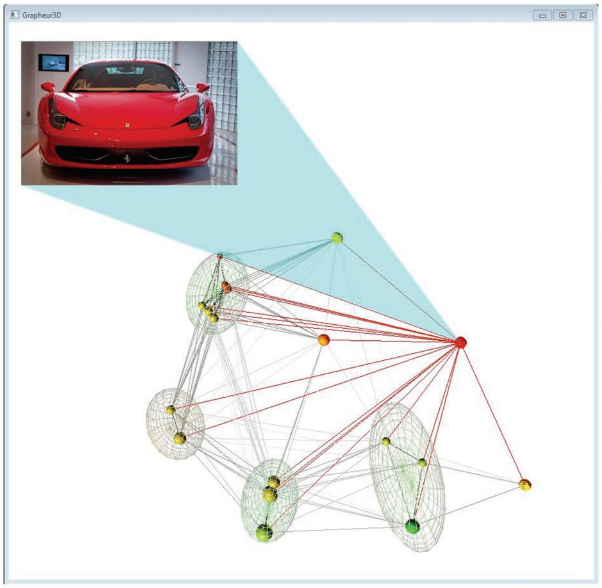


图 16-4 对汽车的机械特性进行聚类

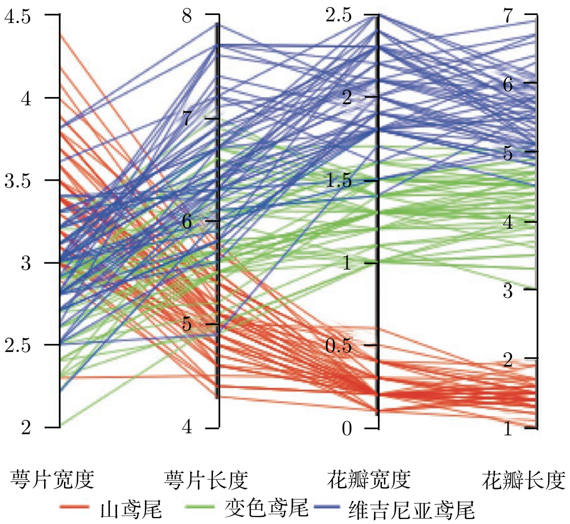


图 16-5 费希尔鸢尾花数据集（每朵花包含 4 个度量属性）的平行坐标展示，每个属性都用一个垂直轴表示，数据中的第 i 项属性值表示为折线与对应的第 i 个垂直轴的交点（另见彩插）

平行坐标图可能是最简单有效却最不为人知的可视化 n 维数据的方法。当 n 大于 2 或 3 时，我们很难直接通过眼睛观察数据。你不必等到成为工程师以后才使用这个方法（现在很多开明的组织已经在使用它）。



梗　概

凝聚聚类生成一棵包含数据点的**树**（**层次结构**）。如果你不熟悉树结构，可以想想用来整理文档的文件夹，无论实际中的还是计算机中的（与某项目相关的文档放在一起，然后与不同项目相关的文件夹合并成一个“工作进行中”文件夹等）。

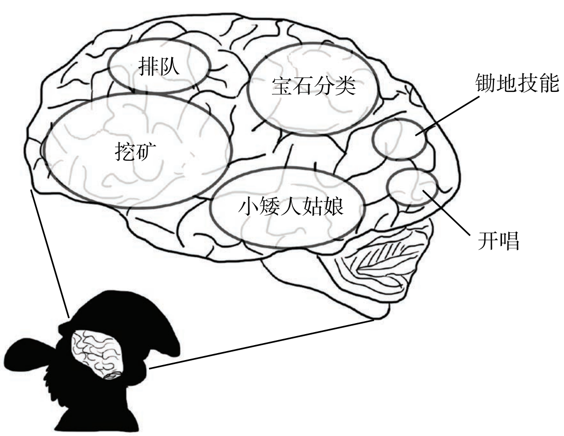
想象一下，你没有秘书，也没有时间手动完成：自底向上的聚类方法可以为你完成工作，只要你找到一个合适的方法来测量单个数据点之间的相似性，以及已经合并的数据点集之间的相似性。

这种方法被称为**自底向上**，是因为它从单个数据点开始，合并最相似的那些点，然后合并最相似的集合，直到获得单一的集合。开始时没有指定聚类的数目，而是用不同的相似性水平来切割这棵树（也称为**树状图**），尝试了若干种不同的切法之后，可以找到一个合适的聚类数目。

通过凝聚聚类，圣诞老人可以把所有的圣诞礼物放在一只很大的红色盒子里。人们打开它后，又发现了一组盒子，再打开，还是盒子　　直到打开真正放着礼物的“叶子”盒子。

第 17 章　自组织映射

> **祖母细胞**是一个假设的神经元，表示任何复杂且具体的概念或对象。当一个人的大脑“看到，听到，或以其他方式判别”一个特定的实体，例如他或她的祖母时，它就会激活。
>
> ——Jerry Lettvin，1969



从前面的章节中，你应该已经熟悉基本的聚类技术了。聚类确定相似数据的分组，有些情况下会用到层次结构（分组，然后组包含组，等等）。如果存在一种内部表示，那么一个组可以用一个原型来表示。本章涉及**原型的排列，排列依据是规则的网格结构，以及这个网格中的邻居的相互影响**。

主要思路是**聚类数据（实体）同时在一个二维图上可视化这个聚类的结构**。人们想要得到的可视化至少应该近似地和聚类相一致——这应该足够使你好奇并愿意继续读下去。

每个聚类 i 都以原型 $\boldsymbol{p}_i$ 作为代表向量。在市场营销领域，常常会标识不同的客户类型，并通过原型（富有的单身汉，已成家的中产阶级工人等）来描述。原型会与我们的实体具有数目相同的坐标，向量的每个分量将描述给定聚类的一个代表值，例如包含在该聚类中的各实体的平均值。

在二维可视化空间里，想要得到连贯的可视化，类似的原型就应该放在邻近的位置。当然，对于高维问题（有两个以上坐标值的问题），没有确切的可用解决方案，因此将目标定为得到足以根据数据进行推理的逼近。**自组织映射**（SOM）是使用无监督学习训练得到的一种人工神经网络，它能产生训练样本的二维表示，称为映射。该模型是由 Teuvo Kohonen 引入人工神经网络的，因此也被称为**Kohonen 映射**。

17.1　将实体映射到原型的人工皮层

自组织映射（SOM）由组件节点或神经元构成。节点的排列是一个二维网格中的正规布局。在某些情况下，网格是六边形的，使得每个节点有 6 个最接近的邻居，而不像传统的方形网格中那样有 4 个邻居（见图 17-3）。每个节点 i 附带一个原型向量 $\boldsymbol{p}_i$ 和映射空间中的位置，其中原型向量与输入数据向量有相同的维数。

再次与我们的神经系统进行类比：**神经元是根据在大脑中连接的物理网络组织起来的，在现实中是二维或三维的**。一些神经元由演化和训练进行调整，当特定事件被触发时，它会放出电信号，如图 17-1 所示。例如，当你母亲进入你的视野时，某个神经元可能会放出信号。这种情况下的原型是通过对应于你母亲的视觉特征给出的，位置则是大脑中神经元的物理位置。

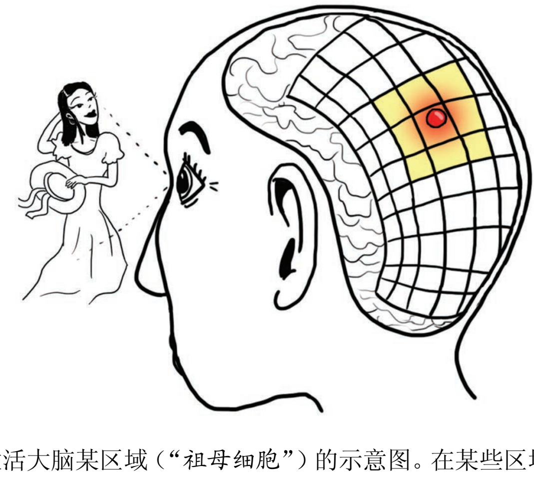


图 17-1　外界刺激激活大脑某区域（“*祖母细胞*”）的示意图。在某些区域，神经元被近似地排列成二维结构，就像大脑的皮层，最高级的功能位于其表层

我们的神经系统的另一个原则是，在许多情况下，**相邻的神经元往往会被类似的输入数据所激发**（你母亲的出现所激发的神经元，其邻居可能会被你母亲的一张老照片所激发）。训练结束后，自组织映射描述从高维度输入空间到二维空间的映射。每个二维的单元对应一个神经元，并且包含一个原型向量。于是一个普通的实体则被映射到（或指定到）原型向量与描

述该实体的向量最接近的神经元，如图 17-2 所示。该训练可以从原型向量的随机初始配置开始（例如选择实体的某个随机子集），然后通过表示和映射随机选择的实例进行迭代。获胜神经元 $c(\boldsymbol{x})$，或简称 c，其原型向量是最接近描述当前实例 $\boldsymbol{x}$ 的向量：

$$c(\boldsymbol{x}) = \arg\min_i \|\boldsymbol{x} - \boldsymbol{p}_i\| \tag{17.1}$$

接下来改变获胜原型 $\boldsymbol{p}_{c(\boldsymbol{x})}$，使之更接近于网络中的当前实例中的那一个。另外，附近的向量的原型也以类似的方式被改变，虽然随着网格中的距离增加，改变的量会越来越小。

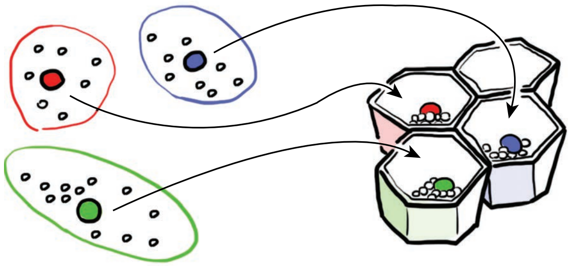


图 17-2　SOM 将多维空间中的实体映射到二维空间的神经细胞中，每个神经元拥有一个原型，并且每个实体都被映射到其最接近的那个神经元中

想想看，在民主制度中，要求选民（实体）教育一组按规则排列的代表（如同在议会中），使他们中至少有一个能代表有相关主张的聚类，坐在附近位置上的代表会相互影响，并且往往会变得相似。存在两个“力场”：实体和原型的吸引力，以及网格中相邻的原型之间的吸引力。实体（选民）争夺原型：每个实体都拉动它的获胜原型，并在较小程度上，获胜原型的邻居朝着自己移动，使之更加相似。当然，不同的实体拉向不同的方向，因此所得到的动力系统是很复杂的。

解释了基本机制和动机后，现在来关注细节。在线学习模式中，每次迭代 t 都提取出来一个随机实体 $\boldsymbol{x}$，确定其获胜神经元 c，并且所有的原型向量 $\boldsymbol{p}_i(t)$ 在迭代（时间）t 时做如下修改：

$$\boldsymbol{p}_i(t+1) \leftarrow \boldsymbol{p}_i(t) \; + \; \eta(t) \cdot \mathrm{Act}\,(c(\boldsymbol{x}), i, \sigma(t)) \cdot (\boldsymbol{x} - \boldsymbol{p}_i(t)) \tag{17.2}$$

其中 $\eta(t)$ 是一个随时间变化的小的学习速率，$\mathrm{Act}\,(c, i, \sigma(t))$ 是一个**激活函数**，它依赖于二维网格中的两个神经元之间的距离，以及随时间变化的半径 $\sigma(t)$。公式中涉及的两个神经元有模式 $\boldsymbol{x}$ 的获胜神经元 c，以及模式 $\boldsymbol{p}_i(t)$ 正在被更新的神经元 i。更新的机制类似于式 (15.9) 中所描述的用于 K 均值软聚类更新的机制，但也有重要的区别：现在的神经元有规则的二维组织，以确定激活电平。

为了帮助收敛，通常学习速率随着时间下降，半径参数也是同样的情况。基本思路是，在开始时，神经元原型移动更快（幼儿的神经可塑性更高），并且往往会激活一大组邻居，然而

到了后来，移动会变慢，并且影响也局限于一小组邻居，这时排列有可能已确定数据分布的主要特征，所需的仅仅是一些微调。某些情况下，学习率随时间递减，比如 $\eta(t) = A/(B+t)$。合理的默认值可以是 $\eta(t) = 1/(20+t)$。

在以批处理的方式训练时，所有 N 个实体 $\boldsymbol{x}_j$ 都呈现给 SOM，确定它们的获胜神经元 $c(\boldsymbol{x}_j)$，然后进行如下更新：

$$\boldsymbol{p}_i(t+1) \leftarrow \frac{\sum_{j=1}^{N} \operatorname{Act}\left(c(\boldsymbol{x}_j), i, \sigma(t)\right) \cdot \boldsymbol{x}_j}{\sum_{j=1}^{N} \operatorname{Act}\left(c(\boldsymbol{x}_j), i, \sigma(t)\right)} \tag{17.3}$$

每个原型都以所有实体的一个加权平均来更新，其中权重正比于神经元网格空间（通常为二维）中获胜神经元的原型和当前原型之间的临近区域。

由于系统的复杂性，建议尝试使用不同的参数和不同的时间安排，直到得到可接受的结果。例如，一个合适的邻域激活函数可以是：

$$\operatorname{Act}\left(c, i, \sigma(t)\right) = \exp\left(-\frac{{d_{ci}}^2}{2\,\sigma^2(t)}\right) \tag{17.4}$$

其中 d_{c_i} 是二维网格中的两个神经元之间的距离，$\sigma(t)$ 是一个邻域半径，一开始它包括的不限于最接近的邻居，不过最后只包括一组临近的邻居。注意，不要混淆网格中的神经元之间的距离（如图 17-3 所示）和数据的原始多维空间原型向量之间的距离！

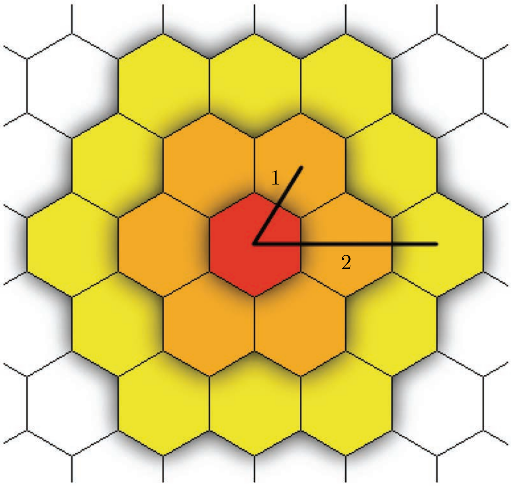


图 17-3　自组织映射中一个邻居的例子：距离分别为 1 和 2 的邻居神经元

令 $\mathrm{TOT_{SOM}}$ 为 SOM 的神经元总数，$\mathrm{TOT_{iter}}$ 为执行迭代的总次数。默认值从 $\sqrt{\mathrm{TOT_{SOM}}}$ 开始，如果网格是方形的，这是一个接近网格半径的值，结束时值为 2，如下所示：

$$\sigma(t) = \frac{(\mathrm{TOT_{iter}} - t)\sqrt{\mathrm{TOT_{SOM}}} + 2t}{\mathrm{TOT_{iter}}} \tag{17.5}$$

该映射或类似映射的复杂本质不应该让用户感到气馁：在许多情况下，基本参数仅考虑简单的默认值就能得到可以接受的结果。但另一方面，这并不奇怪，我们大脑的基本映射机制确实复杂程度很高，因为我们是拥有智能且有一部分变幻莫测的人类，不是吗？

17.2 使用成熟的自组织映射进行分类

即使你不想沉迷于上述数学细节中各种上下标的“灯红酒绿”里，你仍然可以有效地利用 SOM 来指导问题的推理。训练结束后，SOM 可用于为新对象进行分类，先寻找最接近的（获胜）原型，然后将新对象分配到对应的神经元，如图 17-4 所示。在许多情况下，看过原型之后，很容易为不同的神经元*命名*，以帮助推理和记忆。但我们注意到，神经元可能会发现不同寻常的组合，带来有趣的洞见和*新群体的发现*，而不只是重新发现平凡的分类。

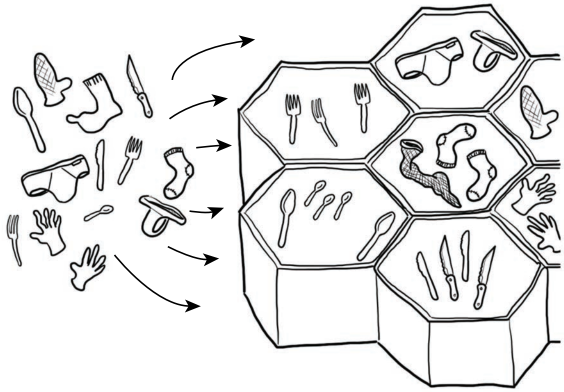


图 17-4 SOM 的一个类比示意图：每个神经元就像橱柜里井然有序的抽屉，相邻的抽屉中存放着相似的物品

想象一下用市场营销数据训练 SOM：每一个神经元可能代表客户的特征组，其名称可以是“富有的单身汉”“有孩子的贫穷家庭”“年长的退休人员”“被宠坏的青少年”等。当一个新客户到来时，你可以很容易地识别出相应的原型，例如选择向他推销你产品的最佳策略。如果你是一个电影迷，训练 SOM 为不同的电影分类，可以使用 SOM 为一个新的电影分类，例如（以很高的概率）预测你是喜欢还是不喜欢。

在 SOM 中，训练的质量可以通过**量化误差**（用获胜原型向量 $\boldsymbol{p}_{c(\boldsymbol{x})}$ 替代实体 $\boldsymbol{x}$ 导致的平均误差，即所有数据向量离其最接近的原型的平均距离）来测量，或通过更复杂的**拓扑误差**，它与赋值相关，在某些情况下，原始高维空间中的接近的向量不能正确赋予神经元网格空间（通常是二维）中接近的神经元。拓扑误差可以用所有数据向量最接近和第二接近的原型（在原多维空间中）在映射中不是邻接神经元的比值来计算。

颜色编码可以用来表示一个维度的数据点的值，而每个六边形的大小可以表示沿着另一维度的值（见图 17-5）。彩色的映射被称为组件或**组件平面**，并且可以通过比较来确定局部关系。可以将 SOM 映射与散点图显示或者平行坐标显示相结合，来制定有趣的新分析技术（见图 16-5）。例如，当鼠标指针移到 SOM 的神经元上，与每个神经元相关联的原型向量的位置可以用散点图显示或以平行坐标显示。以这种方式，可以进一步分析相关实体的细节。

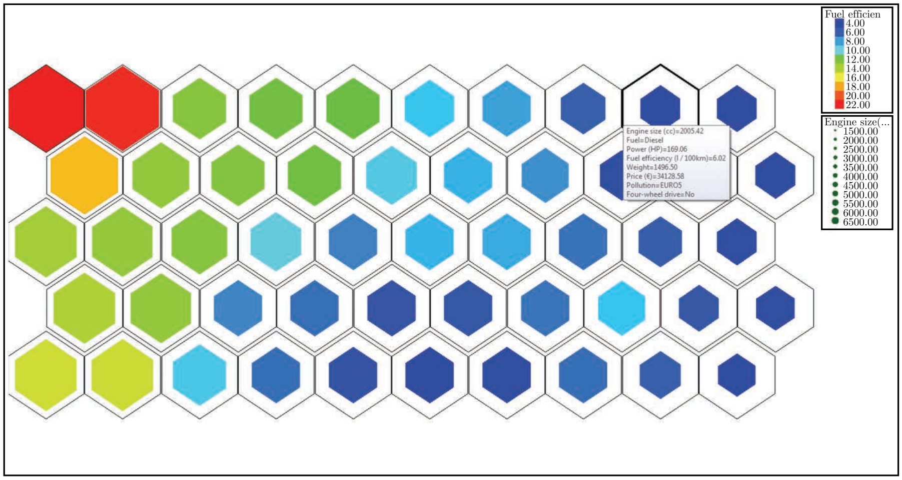


图 17-5　一个 SOM，颜色和大小取决于二维原型向量的两个坐标，可以将鼠标移到神经元上来显示原型的值（通过 LIONoso.org 提供的软件，另见彩插）



梗　　概

自组织映射有两个目标：将一组原型放在接近数据点的聚类旁；让原型以二维网格形式组织，从而让邻近的原型在网格中能经常被映射到类似的数据点。

背后动机部分是生物的（我们的神经皮质大致是由神经细胞的二维和三维结构组织起来的），而另一部分与可视化有关。一种二维网格可以在屏幕上可视化，并且原型的特征不是随机分散的，而是慢慢改变，因为邻居关系会带来更易于理解的可视化效果。

如果将数据点想象成大海里的鱼群，那么 SOM 就是有弹性的渔网，目标是捕捉到最多数量的鱼，又保证网不会破。

第 18 章　通过线性变换降维（投影）

你，享有阴影和光亮，被赋予两只眼睛，拥有透视的能力，陶醉于五颜六色中；你，可以理解角度，可以在三维空间中看到一个圆的完整圆周 —— 我怎样才能向你描述清楚我们在平面大陆中遇见的极度不同？

——《平面国》，1884，埃德温　阿博特



在**探索性数据分析**中，实际上使用的是**大脑的无监督学习能力**，以从数据中找到有趣的模式和关系。将实体映射到二维空间很有用，使得我们用眼睛就可以对它们进行分析。映射必须尽可能多地保留存在于原始数据中的，描述实体之间的**相似性和多样性**的相关信息。例如，想想一个营销部经理分析他的客户之间的相似性和相异性，可以针对不同的人群安排不同的活动；或者人力资源部经理试图把不同员工所拥有的能力进行分类。我们想在二维空间里排列实体，**使类似的实体相互靠近，而不同的实体相互远离**。注意这种方法和 SOM 映射明显不同。在 SOM 中，与二维网格相联系的原型向量被移动（坐标发生改变）以覆盖原始数据空间，而这里的方法将原始数据点通过不同的方式映射到一个二维表面上。

根据第 15 章（见图 15-2）的讨论，回想一下系统中给定初始信息的两种方法。第一种使用实体的可能性是通过**内部结构**（**坐标向量**）表述。这种情况下，必须用原始坐标推导出实体之间的相似性度量，例如考虑两个对应的向量之间的欧几里得距离。第二种使用实体的可能性是通过实体之间**成对关系的外部结构**，通过相似性或相异性来表示。我们只处理相异性以避免混淆，把转变公式就能处理的其他情况留给读者。为了符号表示清晰，让 n 个实体由一些相互之间的相异性 δ_{ij} 来确定。部分相异性可以是未知的。

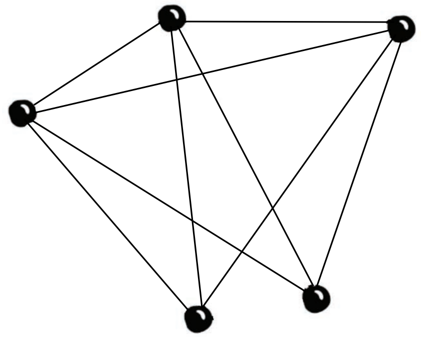


图 18-1　将实体（点）通过外部相异性（边）连接起来

一个合适的模型是无向图，如图 18-1 所示，其中每个实体由一个节点表示，两个节点之间的连接的权重是 δ_{ij}，当且仅当相应实体的距离定义为 δ_{ij}。图中的边的集合表示为 E。

这两种情况（坐标或关系）可以组合起来。在某些情况下，提供给系统的信息包含**坐标和关系两者**。作为一个非常具体的例子，想象一些自动聚类方法被应用到数据向量上。然后我们可以声明两个实体是相异的（$\delta_{ij} = 1$），当且仅当它们不属于同一个聚类。该附加信息可以用来鼓励可视化，其中来自同一聚类的实例往往在二维空间中也很接近。其他情况下，人们给出的实例相异性的指示可以帮助调优可视化，并适应用户的意愿。

用来区分不同上下文的一种方式，需要处理可视化中的监督水平，即过程给定的提示类型。监督的类型包含了从纯粹**无监督的**方法（仅给出坐标）到**监督的**方法（完全给定关系或相异性），再到结合了向量空间中的无监督探索和标记方法的方式。

明确上下文之后，根据可用数据，考虑如何利用这些数据来产生有用的可视化效果。下面的章节中将讨论线性代数得出的方法，而更普遍的非线性方法将在更后面的章节中描述。像往常一样，线性方法简单易懂，非线性的方法原则上更强大但也更为复杂。

18.1　线性投影

本章从线性代数开始。设 n 是向量（实体）的数量，并设 m 是每个向量的维数（坐标数）。为方便起见，这 n 个向量可以作为行向量存储在一个 $n \times m$ 的矩阵 X 里。为了方便读者，拉

丁字母 i 和 j 用来指示数据项，而希腊字母 α 和 β 用来指示坐标。因此，$X_{i\alpha}$ 表示数据 i 的第 α 个坐标。本章的其余部分假定数据是**中心化的**，即整个数据集上的每个坐标的平均值是零：$\Sigma_{i=1}^{n} X_{i\alpha} = 0$。如果原始数据不是中心化的，它们可以通过一个简单的转换来预处理。换句话说，我们对数据点的绝对位置不感兴趣，但是对它们相对于其他数据的相对位置感兴趣。我们用 S 来表示 $m \times m$ 的有偏**协方差矩阵**$S = \frac{1}{n} X^{\mathrm{T}} X$，其中的元素 $S_{\alpha,\beta} = \frac{1}{n} \sum_{i=1}^{n} X_{i\alpha} X_{i\beta}$。

这就是所谓的协方差，因为每一项都测量不同数据情况下两个坐标如何一起变化。若两个坐标同时趋于正值，则协方差中的和将是一个大的正值，而这对于负值来说应该也成立。实际上，如果某个坐标值乘以一个常数（每一次改变物理单位时会发生这种情况，例如，从毫米到千米），协方差将被改变。一个不依赖于物理单位变化的度量是相关系数（correlation coefficient），即协方差除以所涉及坐标的标准差的乘积。（参见 7.2 节。）

我们考虑将这些实例变换到 p 维空间的一个线性变换 L，p 的值通常为 2，但是我们保持更一般的表示。L 由一个 $p \times m$ 的矩阵表示，它以通常的矩阵乘法 $y = Lx$ 作用在向量 x 上。y 的每一个坐标 α 是由 L 的行向量 ν^{α} 和原始坐标向量 x 的标量积得到的。L 的 p 个行向量 $\nu^1, \cdots, \nu^p \in \mathbb{R}^m$ 被称为**方向矢量**，下面我们假定它们具有单位范数 $\|\nu^{\alpha}\| = 1$，因此转化后的 p 维空间的每个坐标 α 都是通过将原始矢量 x **投影**到 ν^{α} 上得到的。如果投影所有实例，并且重复所有坐标，就会得到**坐标向量** $x^1 = X\nu^1, \cdots, x^p = X\nu^p$。

在可能的线性变换中，有趣的可视化是由**正交投影**所得：方向向量 $\nu^1, \cdots, \nu^p$ 相互正交，并具有单位范数 $\nu^{\alpha} \cdot \nu^{\beta} = \delta_{\alpha,\beta}, \alpha, \beta = 1, \cdots, p$，如图 18-2 所示。请注意，这里 $\delta_{\alpha,\beta}$ 是通常的

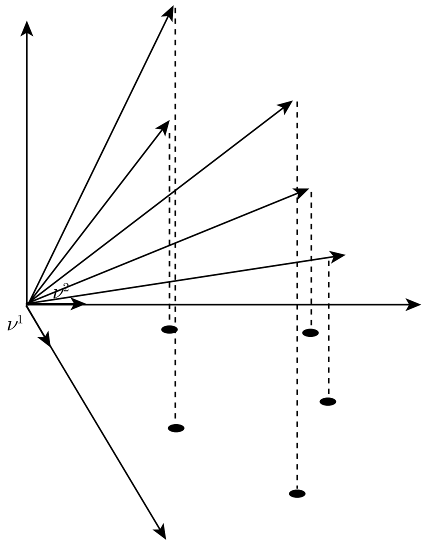


图 18-2　一个投影：每条虚线都连接着一个向量和其在由 ν^1 和 ν^2 定义的平面上的投影（在这种情况下，方向向量是 x 和 y 轴，通常情况下投影可以参照由任何两个不相关向量构成的平面）

克罗内克函数（Kronecker delta），它等于 1（当且仅当两个指标是相等的），不要与相异性混淆！正交投影的一个例子是选择原始坐标的一个子集（这种情况下 $\nu^\alpha = (0,0,\cdots,1,\cdots,0,0)$，所选坐标为 1，所有其他坐标为 0）。其他实例先旋转原始向量，然后选择坐标的一个子集。

这种可视化很简单，因为它显示了数据的**真正性质**，对应原空间中定位的直觉概念，远离数据点，从不同的视角来看数据[①]。想想以任意方位摆放一个二维的屏幕，打开灯（从离数据很远的地方），并观察投影下的阴影。相反，非线性变换可以改变原始数据的分布，以任意的、具有潜在复杂性的、反直觉的方式，就像通过一个变形透镜来观察世界。

作为线性预测的附加功能，它使得解释 p 维坐标很容易，因为每一个坐标都是原来坐标的线性组合（例如，该组合的系数大小"解释"了很多原始坐标与投影之间的关联）。

存储方向向量的存储需求是有限制的，每一点投影的计算复杂度是通常的矩阵向量相乘的复杂度。

现在动机已经有了，接下来考虑一些最成功的线性可视化方法。

18.2 主成分分析

要理解这一历史性的转换（主成分分析由卡尔 皮尔逊于 1901 年发明），就要专注于主成分分析（PCA）尝试解决的问题。像往常一样，*优化是力量之源*，并有助于我们理解运算的深层含义。主成分分析要找到的正交投影，**应能将投影之后的数据元素之间的平方距离的总和**最大化。

如果 dist_{ij}^p 是两个数据点 i 和 j 的投影之间的距离：

$$\text{dist}_{ij}^p = \sqrt{\sum_{\alpha=1}^{p}((X\nu^\alpha)_i - (X\nu^\alpha)_j)^2}$$

PCA 最大化：

$$\sum_{i<j}(\text{dist}_{ij}^p)^2 \tag{18.1}$$

其目的在于使数据点尽可能分散，但只考虑投影则意味着，相互距离不能增加超过原有的距离 $\text{dist}_{ij}^p \leqslant \text{dist}_{ij}$（考虑将毕达哥拉斯定理用于由原始向量、投影向量以及连接投影和原始向量的向量定义的三角形）。我们能得到的最好结果是尽可能近似平方距离的原始总和：

$$\max_{\nu^1,\cdots,\nu^p} \sum_{i<j}(\text{dist}_{ij}^p)^2 \leqslant \sum_{i<j}(\text{dist}_{ij})^2 \tag{18.2}$$

引入这个 $n \times n$**单位拉普拉斯矩阵** L^u 后，由于 $L_{ij}^u = (n \cdot \delta_{ij} - 1)$，优化问题可以写成

$$\max_{\nu^1,\cdots,\nu^p} \quad \sum_{\alpha=1}^{p}(\nu^\alpha)^{\mathrm{T}} X^{\mathrm{T}} L^u X \nu^\alpha$$

①实际上眼睛或相机的映射是一个透视的视角，所以不能仅仅通过字面上的意思来类推。

$$\text{使服从} \quad \nu^\alpha \cdot \nu^\beta = \delta_{\alpha,\beta}, \quad \alpha, \beta = 1, \cdots, p \tag{18.3}$$

通常，**拉普拉斯矩阵**（Laplacian matix）是描述实体间**成对关系**的关键工具。实际上，它在图的学习中应用广泛，两点之间的关系通过连接它们的带权重的边来表示。通常，它是一个 $n \times n$ 对称半正定矩阵，每一行和列之和都为 0。通过拉普拉斯矩阵可以很容易地将**所有成对平方距离的加权和**通过紧凑的方式表示出来：

$$x^{\mathrm{T}} L x = \sum_{i<j} -L_{ij}(x_i - x_j)^2 \tag{18.4}$$

考虑上面介绍的 p 坐标向量，很容易验证：

$$\sum_{\alpha=1}^{p} (x^\alpha)^{\mathrm{T}} L x^\alpha = \sum_{i<j} -L_{ij}(\mathrm{dist}_{ij}^p)^2 \tag{18.5}$$

式 (18.3) 的最优解是由 $m \times m$ 的矩阵 $X^{\mathrm{T}} L^u X$ 的 p 个**最大本征值**（eigenvalue）**对应的本征向量**（eigenvector）给定的。对于中心化的坐标，除去一个乘数的差别（不影响本征向量），该矩阵与协方差矩阵是相同的：$X^{\mathrm{T}} L^u X = n^2 S$。PCA 的解是通过寻找**协方差矩阵的本征向量**得到的。我们在上面的形式中优先使用拉普拉斯矩阵，它可以很容易地泛化到数据点之间的关系存在附加信息的情况。（见 18.3 节。）虽然本节中不给出细节，但是本征向量涉及将问题表述成一个最大化问题：需要最小化的原来的量是二次的，梯度为零以及满足约束条件即可推导出线性（本征值）方程。

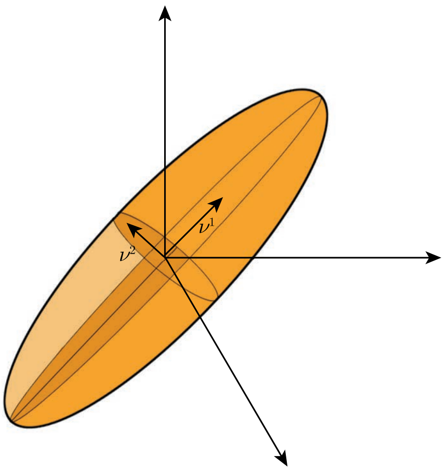


图 18-3　主成分分析，数据点分布在一个椭球体上。第一个本征向量在最长主轴的方向上，即沿着二维椭圆最长的轴，第二个本征向量在与第一个本征向量垂直的平面上

上述解决方案有一个副作用，PCA 将一些可能相关的变量变换为数量较少的不相关的变量，这些不相关的变量称为主成分。第一主成分尽可能多地占据数据中的可变性，并且每个

随后的成分尽可能多地占据其余的可变性。PCA 的另一个有趣的解释是，它最大限度地减少用投影来近似数据时所产生的均方误差。

图 18-3 提供了一个在三维空间中的几何解释。如果我们把数据点想象成 m 维空间中椭球形的云，协方差矩阵的本征向量就是该椭球的主轴。PCA 通过将注意力限制在云的最大分散方向来降低维数。

PCA 是一个简单的、非常受欢迎的转换，但有明显的局限（也许因为太流行了）。它只执行坐标旋转，使得最大方差的方向对准变换后的坐标轴。具有较大方差并不总是意味着具有较多的信息内容，例如，也可能是更大的测量噪声。此外，一个坐标上的方差可以很容易地通过乘以一个常数来增大，而信息内容当然不会改变。换句话说，PCA 的结果取决于是否选择合适的物理单位：即使是把物理距离测量单位由米换成毫米这样简单的原因，也可以让球形的云状点集拉长形状。此外，由于式 (18.1) 的优化涉及平方距离的总和，PCA**对离群值敏感**。与大部分其他点相距很远的点贡献了大的（平方）距离，并使得方向向量的选择可能与离群值被消去的情况有很大的不同。监督学习分类中，当 PCA 用于识别重要的特征时，其主要局限制在于没有利用属性向量的类标签。最大方差的方向并不保证包含有助于区分的好的属性，参见第 7 章有关属性选择和排序的内容。

计算成本与求解维数为 $m \times m$ 的矩阵的特征值–特征向量问题相关。请注意，这与点数 n 是不相关的，因此，当初始坐标数目有限时，这个方法特别快，即使数据点的数量非常大。PCA 的更多细节可以在参考文献[72]中找到。

18.3　加权主成分分析：结合坐标和关系

正如前面所提到的，在某些情况下，数据的附加信息以（部分）实体之间的关系的方式给定。例如，我们可能有一个类标签，使一些点对在同一个类中，我们也希望它们的投影距离比较接近。或者，除了从原始数据坐标中获得的信息，我们还可以有附加的相异性信息。

好在我们可以扩展 PCA 方法来包含更多的信息。例如，可以最小化平方投影距离的**加权和**：

$$\sum_{i<j} d_{ij} \cdot (\mathrm{dist}_{ij}^{p})^2 \tag{18.6}$$

如果权重 d_{ij} 很大，当相应的 dist_{ij}^{p} 也很大时，对将要最大化的函数的贡献也会很大。然后，我们可以将 d_{ij} 解释成衡量点 i 和 j 在低维投影空间里相距很远时的重要性，称为相异性。就像在未加权的情况下，现在可以给这个问题分配一个 $n \times n$ 的拉普拉斯矩阵 L^d：

$$L_{ij}^{d} = \begin{cases} \sum_{j=1}^{n} d_{ij} & \text{如果} i = j \\ -d_{ij} & \text{其他情况} \end{cases} \tag{18.7}$$

而且矩阵 $X^{\mathrm{T}} L^d X$ 的 p 个最高的本征向量给定的方向矢量确定了最优投影。

我们现在可以使用相异值来创建不同版本的 PCA。在归一化的 PCA 中，$d_{ij} = 1/\text{dist}_{i,j}$，由此在优化中大大缩短了原距离。这可以用于提高原始 PCA 针对离群值的健壮性。

在有监督的 PCA 中，数据拍照所属的不同的类来标记。如果 i 和 j 属于同一个类，可以将相异性 d_{ij} 设置为小值 ϵ，反之则设定为 1。权重指导着投影，而投影比让属于不同聚类的点的距离尽可能远要更重要。如果 ϵ 是零，那么每个聚类的内部结构仅间接地根据聚类间成员的关系设定。

18.4 通过比值优化进行线性判别

数量比值的优化带来了考虑类标签的投影数据点的其他可能方法。显然，比值的最大化反映了分子最大化和分母最小化之间的**折中**。

让我们考虑一个 c 向分类问题，标准情况是有两个输出类。费希尔分析找到一个向量 ν_F，当原始矢量投影到该向量上时，不同类别的值会以最佳的方式分离开。

如果投影点的平均散度（scatter）归一化，而投影点的样本均值尽可能不同，可以得到一个很好的分离。通过散度来区分对应于这一直觉，即重要的不是均值本身的分离，而在于：如果数据值都离它们的均值足够近，那么它们的类可以被清楚地分开。如果数据分散使得大多数值被混合在同一区域，即使均值分离，分类也几乎不可能。

令 n_i 为第 i 类中点的数量，令 μ_i 和 S_i 分别为均值向量和第 i 个聚类的有偏协方差矩阵。矩阵 $S_{\text{within}} = \frac{1}{n}\sum_{i=1}^{c} n_i S_i$ 是**类内平均协方差矩阵**，而矩阵 $S_{\text{between}} = \frac{1}{n}\sum_{i=1}^{c} n_i \mu_i \mu_i^{\mathrm{T}}$ 是**类间平均协方差矩阵**。

从细节上来说，费希尔线性判别式被定义为线性函数 $y = \nu^{\mathrm{T}}\boldsymbol{x}$，它最大化下面的这个比值：

$$\frac{\nu^{\mathrm{T}} S_{\text{between}} \nu}{\nu^{\mathrm{T}} S_{\text{within}} \nu} \tag{18.8}$$

考虑**最大化类间与类内散度的比值**：我们希望最大程度地分离各个类（比值中分子的作用，其中依靠均值的投影），并保持聚类尽可能紧凑（比值中分母的作用）。

可以证明，费希尔准则的最大化对应着下式的最大化：

$$\frac{\nu^{\mathrm{T}} S_{\text{between}} \nu}{\nu^{\mathrm{T}} S \nu} \tag{18.9}$$

一个特殊而有趣的二分类例子见图 18-4，特殊化上述方程之后，费希尔线性判别式被定义为线性函数 $y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$，它最大化下面的准则函数：

$$\text{分离}(\boldsymbol{w}) = \frac{\|\widetilde{m}_1 - \widetilde{m}_2\|^2}{\widetilde{s}_1^2 + \widetilde{s}_2^2} \tag{18.10}$$

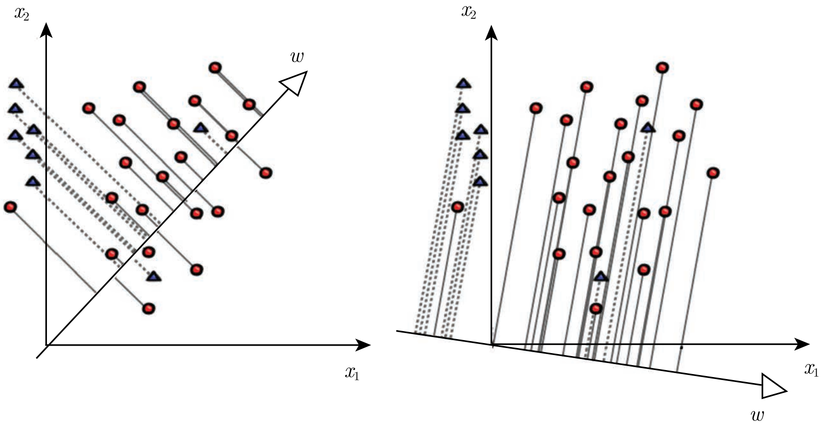


图 18-4　费希尔线性分类模型（三角形表示一个类，圆形表示另一个类）：左边的一维投影将两个类混合在一块了，而右边的投影通过投影样本散度可以最佳地分开这两个类的投影点

其中 $\widetilde{m}_i$ 是投影点的样本均值 $\widetilde{m}_i = (1/n_i)\sum_{y\in \text{Class}_i} y$，$\widetilde{s}_i^2$ 是每个类投影样本的散度：$\widetilde{s}_i^2 = \sum_{y\in \text{Class}_i}(y-\widetilde{m}_i)^2$。考虑最大化类间与类内总散度的比值，解是：

$$\boldsymbol{w}_F = (S_w)^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2) \tag{18.11}$$

其中 $\boldsymbol{m}_i$ 是第 i 类的 d 维样本平均值，S_w 是两个散度矩阵 S_i 的和，S_i 定义如下：

$$S_i = \sum_{\boldsymbol{x}\in \text{Class}_i} (\boldsymbol{x}_i - \boldsymbol{m}_i)(\boldsymbol{x}_i - \boldsymbol{m}_i)^{\mathrm{T}} \tag{18.12}$$

费希尔线性预测的一个有趣的应用是用于神经网络的特征选择和基于监督学习的一般模型建立技术。（参见下一节“用于特征选择的费希尔判别指标”。）

用于特征选择的费希尔判别指标

下面来考虑一个二分类问题（有两个输出的类），有 d 维输入向量。费希尔分析寻找向量 $\boldsymbol{w}_F$，当原来的矢量投影到它上面时，两个类的值以最佳的方式被分离开。该方法已在 18.4 节中介绍过。

让我们回忆一下，如果以投影点的平均散度来归一化，而投影点的样本均值尽可能不同，就可以得到一个很好的分离。通过散度来区分对应于这一直觉，重要的不是均值本身的分离，而在于：如果数据值都离它们的均值足够近，它们的类就可以被清楚地分开。如果数据分散使得大多数值被混合在同一区域，即使均值分离，分类也几乎不可能。

现在可以根据式 (18.11) 及式 (18.12) 中定义的费希尔向量 $\boldsymbol{w}_F$ 的第 i 个分量的大小来评判特征的重要性。识别费希尔向量最大的组成部分将启发式地确定用于分类的最相关的方向

（坐标）。换句话说，如果一个坐标向量的方向与费希尔向量方向相似，那么投射到给定的坐标轴可以近似地用于分类，而不是投影到原来的费希尔向量。注意，该准则是经验上的而不是理论上的，因为它是基于线性投影的：在某些情况下，上述费希尔准则排名靠后的特征的非线性组合在分类中可以做得很出色。

如果维数 d 非常大，式 (18.11) 中的矩阵 S_w 的逆的数值求解可能有困难，第一个度量可能不足以为这许多特征正确排序。

一种更简单但可能更有效的为特征 k 排名的判据称作**费希尔判别指标**，考虑沿特定方向 k 的向量 $\boldsymbol{e}_k$（处处为零，只有第 k 个坐标为 1），它测量式 (18.10) 中定义的"分离 ($\boldsymbol{w}$)"值。换句话说，我们想测量只考虑给定点第 k 个坐标的判别。

18.5 费希尔线性判别分析

上述原始费希尔法旨在找到单一的方向向量（单个投影）。为了找到 p 个方向向量，这个想法可以推广到费希尔**线性判别分析**（LDA）这一常用技术，它基于下述比值的最大化：

$$\max_{\nu^1,\cdots,\nu^p} \quad \frac{\sum_{\alpha=1}^{p}(\nu^\alpha)^{\mathrm{T}} S_{\text{between}} \nu^\alpha}{\sum_{\alpha=1}^{p}(\nu^\alpha)^{\mathrm{T}} S \nu^\alpha} \tag{18.13}$$
$$\text{使服从} \quad (\nu^\alpha)^{\mathrm{T}} S \nu^\beta = \delta_{\alpha\beta}, \quad \alpha,\beta = 1,\cdots,p$$

尽管 LDA 使用广泛，但就像基本的 PCA，它对离群值很敏感，没有将类的形状和大小考虑在内。更灵活的推广是基于最大化以下形式的比值：

$$\max_{\nu^1,\cdots,\nu^p} \quad \frac{\sum_{i<j} d_{ij}(\text{dist}_{ij}^p)^2}{\sum_{i<j} \text{sim}_{ij}(\text{dist}_{ij}^p)^2}$$
$$\text{使服从} \quad (\nu^\alpha)^{\mathrm{T}} X^t L^s X \nu^\beta = \delta_{\alpha\beta}, \quad \alpha,\beta = 1,\cdots,p$$

其中 d_{ij} 是相异性的权重，sim_{ij} 是相似性的权重（它们表示投影中两个实体在一起的倾向），而 L^s 是对应于如下相似性的拉普拉斯矩阵

$$L_{ij}^s = \begin{cases} \sum_{j=1}^{n} \text{sim}_{ij} & i = j \\ -\text{sim}_{ij} & i \neq j \end{cases} \tag{18.14}$$

将 (A,B) 的广义本征向量问题定义为 $Ax = \lambda Bx$ 的解，式 (18.13) 的最优解通过 $(X^{\mathrm{T}}L^dX$, $X^{\mathrm{T}}L^sX)$ 的 p 个最高广义本征向量给出。

除了数学细节，请务必记住，找到一个最佳的投影需要以定量的方式来定义什么是最优。之前我们已经看到**无监督方式（仅基于坐标）和监督信息（基于关系）的组合**，以及将实例放得近或放得远，不同偏好、不同权重的方法。

当用户把聪明才智用在定义优化问题上时，剩下的就是由相应的乘法运算来推导一个 $m \times m$ 的矩阵，以及使用高效的、数值上稳定的方式来解决一个 $m \times m$ 的矩阵的广义本征向量的问题。当然，若原始坐标数 m 是有限的，则该技术是非常快的，即使要投影的点的数量非常大。

有趣的是，我们将在有关 Web 挖掘的第 25 章中再次遇到本征向量，用于网页排名。



梗　　概

可视化（抽象数据的可视化表示）辅助人们的无监督学习能力，以从数据中获取知识。由于可视化是为我们的视觉系统设计的，它们受限于我们视网膜上的两个维度（如果是立体视觉，就是三个维度）。

一个将数据转换成二维景象的简单方法是通过投影。（实际上，如果是由一台计算机来使用投影点，投影可以多于两个或三个维度。）**正交投影**可以直观地解释为从不同距离来观察数据。

由于有数不清的方法来投影数据，优化就派上用场了，通过明确的目标来选择其中一些方法。特别是**主成分分析**（PCA）确定一个正交投影，它使得投影的点在投影平面上尽可能分散。尽管 PCA 很受欢迎，但它可能无法给出相关的见解：具有较大方差并不总是意味着具有最多的信息内容，或最好的分割。

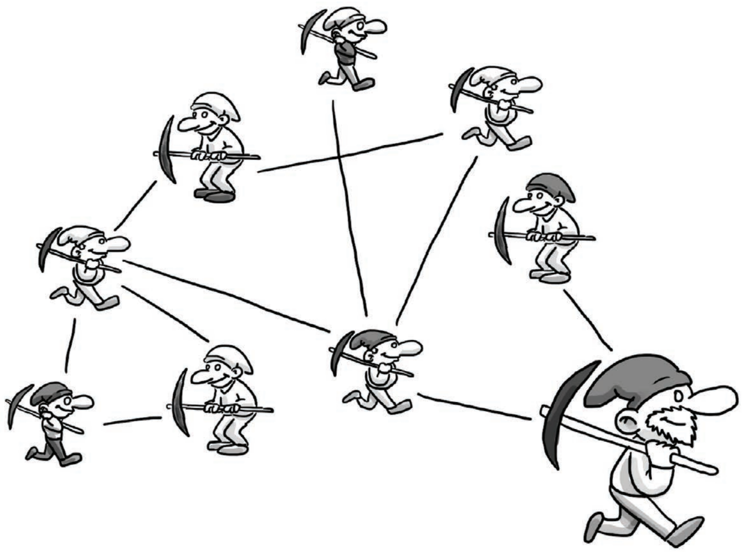
除了原始坐标之外，如果**相互关系**也是已知的（例如，知道某些点在相同或不同类），它们可以用于修改 PCA，以获得更加有意义的投影。

当类标签都是已知的，**费希尔判别**进行数据投影，使得不同类别的投影均值的差与类内散度的比值最大化。

炼金术士使用投影来混合粉状贤者之石与熔融的基本金属，使它们变成黄金。你可以用投影来使企业更加成功，将原始数据点转换为珍贵的洞见。

第 19 章　通过非线性映射可视化图与网络

> 没有人是一座孤岛，可以自全 …… 任何人的死亡都是我的损失，因为我是人类的一员；因此不要问丧钟为谁而鸣；它就为你而鸣。
>
> —— 约翰　多恩，1623



第 18 章介绍了基于线性投影的可视化，现在考虑提高无监督学习能力和从数据中获取洞见的更一般的方法。假定要展示的 n 个实体不一定由内部坐标表示，而是仅由两个实体 i 与 j 的项与项之间（即外部的）**关系表示**，例如相异性 d_{ij}。如果实体确实有坐标，外部关系可以通过如 15.2 节中所解释的简单的方法来获得。

然而，在一般情况下，外部相异性度量不能作为距离来计算，并且可能无法为每一对实体都提供度量。针对这种情况，一个合适的模型是**无向加权图** $G(V,E)$，由一组顶点（或节点）V 和边 $E \subset V \times V$ 确定。每个实体由一个节点表示，并且两个节点之间的连接 (i,j) 被标记为 d_{ij}，当且仅当相应实体间的相异性被定义，如图 18-1 所示。我们假设相似性为正，但不考虑任何其他的假设（如三角不等式）。例如，市场上两种产品之间的相似性，可以通过对客户进行抽样，并要求他们在给定分值范围内来评估产品相似性而得到。

19.1　最小应力可视化

根据眼睛的工作方式，可视化仅存在于二维或三维中。因此，我们的目的是将实体放在二维平面上（或三维空间中），使得**它们之间的相互距离尽可能接近其相异性**。在一般情况下，一个服从所有相异性的完美放置是不可能的。因此，为了确定哪些放置是可以接受的，需要一个明确的标准。

因此，接下来的问题是：给定一组实体之间的（正）相异性 d_{ij}，找到所有实体的二维或三维的坐标 $\boldsymbol{p}_i$，使得实体便于放置，并尽可能保持原始相异性。最简单的方法是**应力最小化**，应力是相对于原相异性由可视化引起的相异性压缩或拉伸。这一方法是直观的、物理的和实用的，也是了解其他更为复杂的方法的起点。

一个直接的误差度量可以量化为平面上的距离相对于原来的相异性有多大差距。为简单起见，下面来考虑二维可视化。

令 $\delta_{ij}=\sqrt{(\boldsymbol{p}_i-\boldsymbol{p}_j)^{\mathrm{T}}(\boldsymbol{p}_i-\boldsymbol{p}_j)}$ 为实体 i 和 j 在平面上的坐标之间的距离。一个自然的*全局映射误差*可以定义为个体平方误差的总和：

$$\sum_{(i,j)\in E}(d_{ij}-\delta_{ij})^2$$

缺失的边（相应的点对没有赋予相异性的值）对误差没有贡献。可以通过添加任意的权重 w_{ij} 来获得额外的灵活性，权重 w_{ij} 表示个体误差对总体应力的影响：

$$\text{全局映射误差}=\text{应力}=\sum_{(i,j)\in E}w_{ij}(d_{ij}-\delta_{ij})^2 \tag{19.1}$$

例如，若 $w_{ij}=1/d_{ij}^2$，人们考虑的则是*相对*误差 $(\delta_{ij}-d_{ij})/d_{ij}$，而不是绝对误差。值 $w_{ij}=1$ 是默认值。

一个精确的解决方案能重现所有原始相异性 $\delta_{ij}=d_{ij}$，并获得零误差。低误差值意味着许多距离往往与原始距离相当接近。换言之，现在这个问题是通过改变点的位置 $\boldsymbol{p}_i$，以最小化*全局映射误差*测量。我们有在二维中放置每个点的完全自由，导致优化问题的维数非常大，等于实体数目的两倍。现在的情况与第 18 章不同，当时我们用一个线性投影进行映射。

最小化以上*全局映射误差*有一个相关的物理模型，这也解释了目前为何广泛使用术语*应力*来称呼被最小化的函数。每对点之间由一个弹簧连接，松弛时长度等于 d_{ij}，也就是所期望的距离，弹性常数（抗形变能力）等于权重 w_{ij}。项 $w_{ij}(\delta_{ij}-d_{ij})^2$ 可以认为是弹簧相对于松弛长度被拉长或压缩而形成的势能。各点的初始位置可以随机选取，而且移动被约束在二维中。系统将开始振荡，如果存在一些摩擦，振荡将逐渐衰减，从而使状态趋于稳定，整体的应力函数达到一个局部最优点。

当然，物理系统可以用一台计算机模拟，从而形成所谓**力控制的绘制图形方式**。基于这种方式的方法包括两个主要组件。第一个是**量化绘图**（或二维映射，如果你喜欢更技术化的

术语）质量的模型。第二个是基于该模型的用于计算局部最优图形的**优化方法**。最终得到的布局使系统达到平衡，因此每个顶点的合力为零，或等价地，顶点的位置使势能达到局部极小值。

如果你不喜欢物理，但是喜欢数学，你可以忘掉摩擦力这样的物理细节，而专注于通过梯度下降法来最小化应力函数：计算偏导数，使用优化方法，达到全局最优解。再说一次，优化是力量之源！

图 19-1 是可视化的例子，它展示了对不同类型的登山活动感兴趣的朋友的**社交网络**。图 19-2 展示了一个议员的社交网络，根据他们议会活动的相似性自动可视化。注意主要政治团



图 19-1　应力最小化的二维可视化

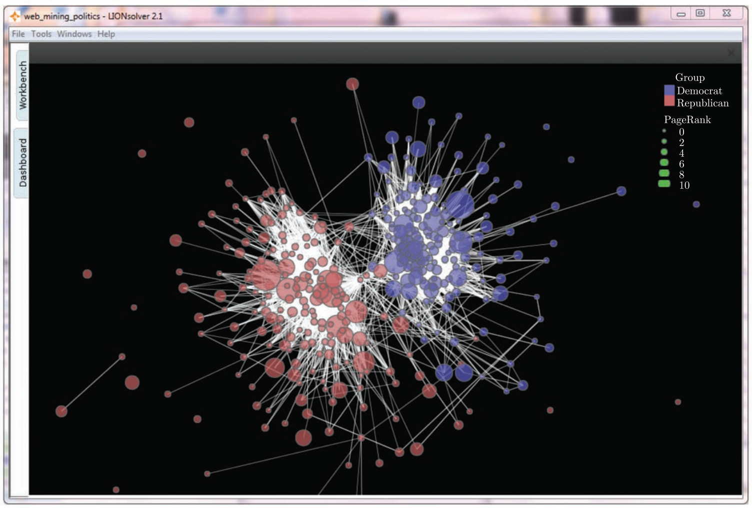


图 19-2　社交网络分析：美国议员的可视化网络。两个政党（从聚类软件无法得到）呈现出非常不同的两个类别（另见彩插）

体是如何自动聚类的，这是鼓励将相似的人放在相似位置的一个有趣的副作用。通过一个**焦点和上下文的可视化**，可以关注一个政治家（焦点）周围的局部连接网络，也可以看全部连接（见图 19-3）给出的情境。因此，很容易就能从实体导航到邻居，再到邻居的邻居等，以同样快速有效的方式来追踪复杂的关系。这种方法的另一种可能的应用是犯罪侦查。

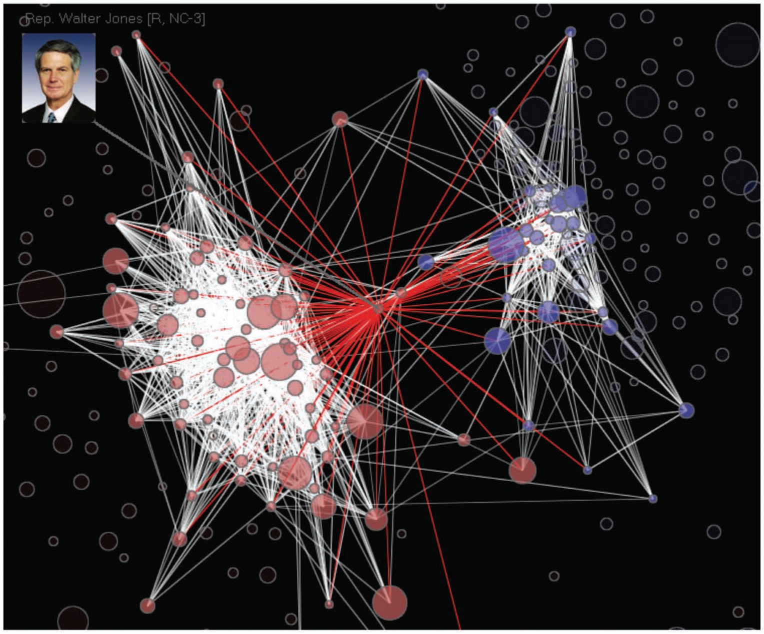


图 19-3　浏览政治社交网络：一个政治家的网络

19.2　一维情况：谱图绘制

一个典型的例子是，将一组 n 个点映射到一维，同时保持相似的点之间尽可能靠近。像往常一样，需要定义想要最小化的那个量，这涉及一维映射的优度。令 x_i 为分配给点 i 的一维坐标（让 $\boldsymbol{x}$ 表示所有这类的坐标向量）。20 世纪 70 年代首次提出的**霍尔能量**（Hall's energy），计算式如下：

$$E_{\text{Hall}} = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(x_i - x_j)^2 \tag{19.2}$$

此式的解释是将各个距离求平方（使得该函数可微，并且该微分可得出线性方程），并以点对之间的相似性为权重将这些平方值求和。当 w_{ij} 很大时，$(x_i - x_j)^2$ 这一项对函数 E_{Hall} 有很大的贡献，因此该定义鼓励把类似点放在一起，以避免严厉的惩罚，即大的 E_{Hall} 值。通过霍尔能量，相似性高的点对就能被放置在相近的地方。

现在暂停一下，看看上述定义和解释中的一个严重的缺点。现在可以完全自由地为每个点选择一个坐标 x_i，这样就能通过选取非常小的坐标（或非常相似的坐标）使能量趋近于零，但只给我们留下了一个平凡解：所有点都映射到相同的位置。这个定义可以修复，因为我们感兴趣的不是坐标的绝对值，而是相对值。通常优化的绘图不应该依赖于选择米或毫米为单位。因此，我们可以将 x 向量的长度固定为 1，于是问题变为：

$$\text{最小化}\quad \left(\sum_{(i,j)\in E} w_{ij}(x_i - x_j)^2\right) \tag{19.3}$$

$$\text{使服从}\quad \|\boldsymbol{x}\|^2 = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = \sum_{i=1}^{n} x_i^2 = 1 \tag{19.4}$$

为了方便，令 $N(i)=\{j|(i,j)\in E\}$ 为节点 i 的邻居，$\deg(i)=\sum_{j\in N(i)} w_{ij}$ 为加权度。一旦定义了一个图相关的拉普拉斯矩阵 L^G：

$$L_{ij}^G = \begin{cases} \deg(i) & i = j \\ -w_{ij} & i \neq j \end{cases} \tag{19.5}$$

就可以得到霍尔能量 $E_{\text{Hall}} = \boldsymbol{x}^{\mathrm{T}} L^G \boldsymbol{x}$。

能量和约束具有平移不变性。我们可以消除这个自由度，即令 $\boldsymbol{x}$ 的均值为零：$\sum_{i=1}^{n} x_i = \boldsymbol{x}^{\mathrm{T}}\mathbf{1}_n = 0$（其中向量 $\mathbf{1}_n$ 是全部为 1 的向量）。最后，一维优化分布可以用以下带约束最小化问题的解来描述：

$$\text{最小化}\quad \boldsymbol{x}^{\mathrm{T}} L^G \boldsymbol{x}$$

$$\text{使服从}\quad \begin{cases} \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = 1 \\ \boldsymbol{x}^{\mathrm{T}}\mathbf{1}_n = 0 \end{cases}$$

通过标准的优化和线性代数的结果，只要图是连通的，所得能量的最小值就是所谓的*代数连通度*，即 L^G 的第二小的本征值 λ_1（L^G 是奇异矩阵，因此其最小本征值 $\lambda_0 = 0$），同时其解为对应的本征向量 $\boldsymbol{v}_1$，也称为*费德勒向量*（Fiedler vector）。这一结果是优雅的，值得拥有一个鼓舞人心的名称：**谱图绘制**，或**光谱分布**。术语“谱”（spectral）与幽灵和恐怖电影无关 ①，而与物理学中的本征向量和本征值相关，它们用于研究由一个辐射源（频谱）发射的能量和振动模式的分布等。

但对于一维以上的空间，就做不到优雅了。让我们称第二个维度为 y。一个平凡的推广是使得第二个向量坐标 $\boldsymbol{y}$ 与 $\boldsymbol{x}$ 相同，但不是很有收获：所有点都会排列在对角线上，而不是一个真正的二维图。为了得到更有用的结果，我们必须强制解决方案中的 y 坐标与 x 坐标不同。

一个合理的要求是令这两个坐标向量不相关（$\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} = 0$），从而使附加的维数能给我们一些新信息，“新”是指它不线性相关先前的值，不是某个深层的信息论含义。现在关于 $\boldsymbol{y}$ 的问

① spectral 也有“幽灵般的”“鬼魅的”“无形的”的意思。—— 译者注

题就变成了：

$$\text{最小化} \quad \boldsymbol{y}^{\mathrm{T}} L^{G} \boldsymbol{y}$$

$$\text{使服从} \quad \begin{cases} \boldsymbol{y}^{\mathrm{T}} \boldsymbol{y} = 1 \\ \boldsymbol{y}^{\mathrm{T}} \mathbf{1}_n = 0 \\ \boldsymbol{y}^{\mathrm{T}} \boldsymbol{x} = 0 \end{cases}$$

潜在的困难是如何解决非常大的本征向量问题。用于计算主本征向量的多标度技术和迭代技术可以提供帮助。

尽管以著名的线性代数结果来最小化简单函数的做法很优雅，但这不能保证分布的美观性对应用户的喜好。尤其是该方法有可能将太多节点靠得太近，以至于几乎看不清，因为这方面没有规定禁止。此外，也无法保证 y 坐标对应最佳的美学效果，因其不相关。

现实世界中的分布通常需要能量（需要最小化的函数），设计成与特定的偏好相吻合。通过定义一个明确的能量，人们可以将对**目标**（所需的布局特征）的关注从对**如何达到目标**的关注中分离出去。通过优化技术，我们至少可以得到近似的目标。

19.3　复杂图形分布标准

考虑下面的简单图连接矩阵，其中两个节点 i 和 j 连接，当且仅当矩阵项 (i,j) 为 1：

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

该矩阵对应于一个有 3 个节点的图，其唯一要求是，节点 2 到另外两个节点距离都是 1：

$$d_{12} = d_{23} = 1$$

从应力最小化方法的角度来说，图 19-4 所示的分布都是完全等效的：若边不存在（例如，节点 1 和节点 3 之间）则表明，如果需要进行优化的能量函数仅包含有关连接的项，相应节点之间的相互距离是无关紧要的。

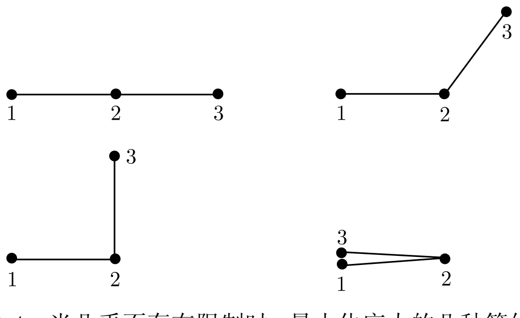


图 19-4　当几乎不存在限制时，最小化应力的几种等价形式

当一个大图存在许多无关紧要的节点对时，情况就会更糟。图 19-5 显示了一个 30×30 的矩形点阵，其中只定义了最邻近的边（要求节点之间的距离是单位距离）。图 19-6 显示了通过最小化式 (19.1) 获得的“最优”分布。许多退化的局部最优分布是存在的。它们可以通过类似棋盘的方式，交替使用黑色和白色节点来获得，使黑色节点仅连接到白色节点，反之亦然。一个一维的解决方案，是将所有黑色节点放在 $x = 0$，所有白色节点放在 $x = 1$，这样就能满足所有的距离限制，使式 (19.1) 中定义的全局映射误差为零。

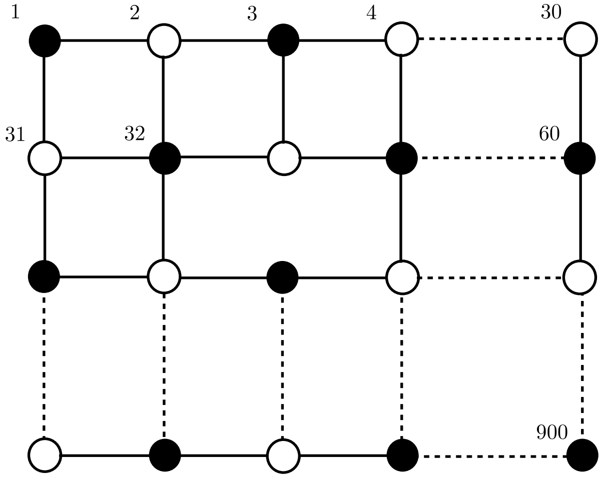


图 19-5　一个定义最邻近边的 30×30 点阵

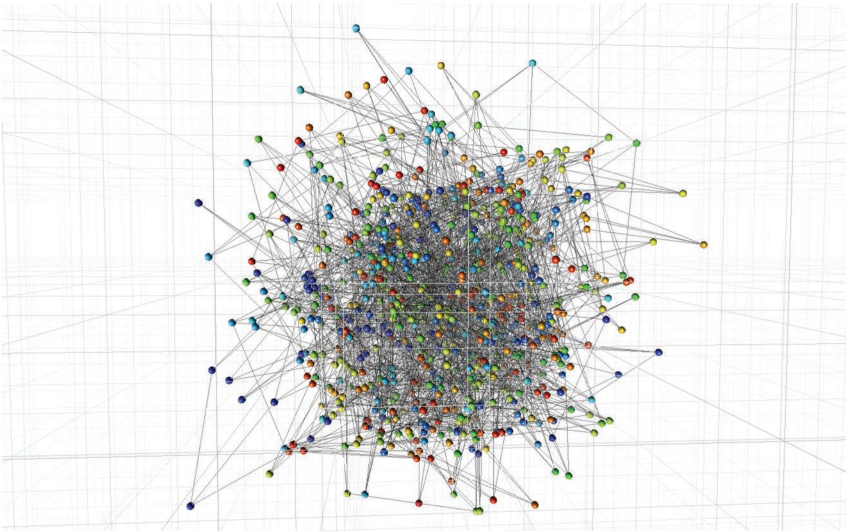


图 19-6　一个最小化应力视角下的“最优”分布，但不是理解网络结构的最优形式

通过为未连接的节点引入一个默认的较大距离，可以轻松解决这个问题。图 19-7 显示了在同样的 30×30 点阵上的最优分布，它是由式 (19.1) 获得的，其中断开的节点 i 和 j 需要较大的距离 $d_{ij} = 20$，还有非常小的权重 $w_{ij} = 10^{-5}$。可以观察到，由于该分布先验上是未知的，常常很难确定一个合适的默认距离。例如图 19-7 中，对于保持正确的分布，20 这一值过小，它会使整个图形弯曲成球形。

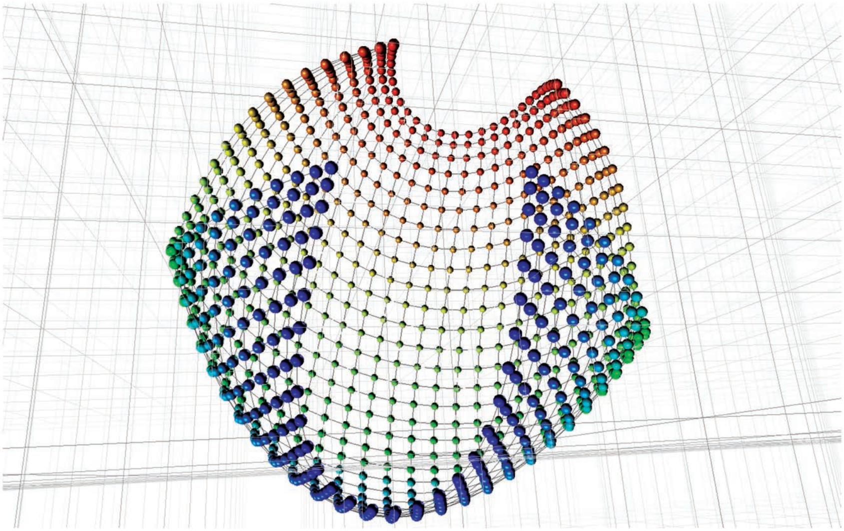


图 19-7　限制缺失的第一种解决方案：引入默认的互斥力

第二种方法，如图 19-8 所示，是通过**最短路径**的计算补充距离矩阵。所有非直接连接的节点间的距离设置为等价于节点之间的**最短路径**长度。例如图 19-5 所示的点阵中，节点 1 和 32 之间有一个最短路径，长度等于 2（一横边加一纵边）。如果没有复制最短路径距离的要求，就没有什么能禁止节点 1 和节点 32 在可视化中被放置得非常近。一旦这个要求被激活，这种不好的行为会受到严厉的惩罚，节点则倾向于分散，从图 19-6 的结构变成图 19-8 的结构。

注意，该最小路径距离总是比网格分布中的欧几里得距离要大。作为一个例子，在两个对角极端节点 1 和节点 900 之间的最短路径的距离为 $(30-1)\cdot 2 = 58$，而在网格分布中欧几里得距离的期望是 $(30-1)\cdot\sqrt{2} \approx 41.01$，因此图 19-8 是个枕形分布。

根据**额外的审美标准**，可以最小化更复杂的函数生成不同结构的图，比如最小化交叉边的个数、保证连接一个点的边的最小夹角（小夹角的可读性差）、或者允许曲边的存在等方案，本章就不在此罗列了。在所有情况下，当定量地定义好一个平衡各种理想美学标准的合适的折中方案后，我们就可以寻找一个有效的最小化算法（力量的来源），在很多情况下意味着寻找一个近似但高效的算法。

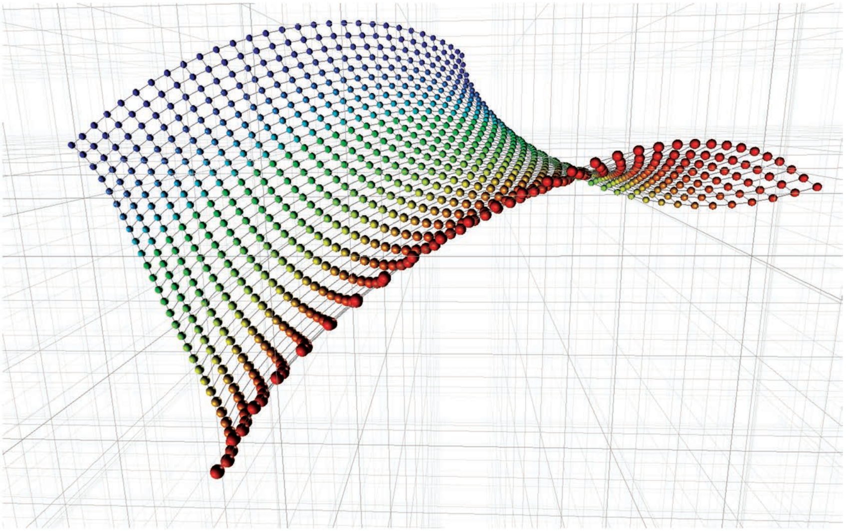


图 19-8　限制缺失的第二种解决方案：通过计算最短路径补充距离矩阵



梗　　概

图形分布技术可用于可视化实体之间的关系。

如果相异性是已知的，在二维空间里绘制实体，使得类似的项彼此接近，对于识别组（聚类）和组间的关系是重要的。

应力最小化诉诸某种物理模型。每个相异性值为 n 维实体之间造了一根弹簧。目标是通过挤压将网络“夹入”到一个平面上，同时最小化各弹簧的伸长或缩短程度。顺便说一下，如果弹簧被刚性杆取代，挤压就会变得不可能：一般来说，不存在点映射到平面且保持所有相异性值不变的精确解。如果你将每个点想象成参加聚会的人，那么每个人都在地板上移动，远离讨厌的人，靠近喜欢的人。每个人同时都在移动，可能会让聚会（可视化）变得非常紧张（次优）。

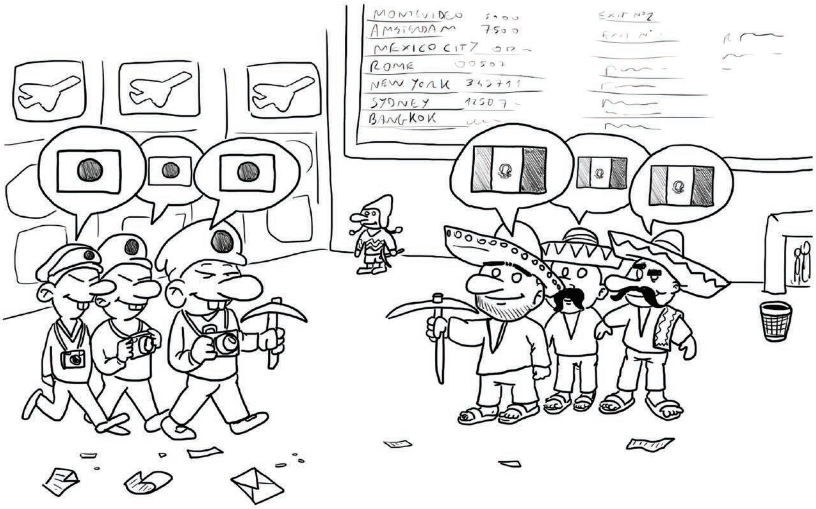
对于聚类，没有绝对最优的图（或网络）分布。通过优化定义**目标**（“最优分布”的定量含义）的一个函数，然后确定最大化它们的最可能的映射。在确定合适的可视化方案之前，人们经常尝试多种可能性。

社交网络分析被用来研究相互作用的人之间的网络。在企业中，员工之间的相似性可以通过他们相互收发信息的数量来确定。如果你用这个指标来设计员工网络的分布，你很容易就能识别在一起工作的同事的聚类：不同群体之间的连接可能会稀疏一些，而有些看似不合群的人，也许非常专注，也许更喜欢打电话，也许　　对工作不太上心吧。

第 20 章　半监督学习

心灵是用来点燃的火，而不是一艘用来装满的船。

—— 普卢塔克



考虑第 17 章中无监督学习提到的国际机场的例子：你走过一个登机口，注意到说着不同语言的人们分别聚集在一起，即使你不知道语言的名称。如果现在能够确定某些语言，比如有人挥舞着国旗或者穿着他们国家的传统服饰，那么我们可以只选择那些已标记的人，然后运行监督学习算法将语音特性映射到语言。

现在的问题就是：是否可以使用未标记实例中的信息来改进语言归类？我们注意到聚在一起的人们往往说同一种语言（“羽毛相同的鸟聚集在一起”），并且如果同一聚类中的至少一个成员说这种语言，可以尝试以相同的语言来标记一些语言未知的人。如果这一假设为真，将大大增加实例的数目，并且可以提高训练的分类器的整体泛化能力。例如，幼儿与他们的年长的已确定语言标记的父母聚类在一起，并且可以被添加到数据库中，这样即使是年轻人的声音（通常频率更高）也可以被正确地分类。

以类似的方式，人们可以使用一些有标记数据，来协助无监督学习和聚类。这是半监督学习的基本思想：**使用已标记的实例，以及（一些）未标记的实例，以提高整体的分类准确率。**

如果假设有效，那么对于**已标记实例稀缺和未标记实例丰富**的所有情况，会得到一个非常有价值的性能提升。想想网页上的例子：人工制作的标签是非常昂贵的，只有很小一部分网页被标记。相反，有大量的未标记网页，并且数量还在不断增长。

20.1 用部分无监督数据进行学习

半监督学习（semi-supervised learning，SSL）同时使用监督和无监督的数据，以提高性能。监督的标准形式是一些实例上的**标记**。在这种情况下，训练集 X 被分成已标记的部分 $X_L = \{x_1, \cdots, x_l\}$（它们的标记 $Y_L = \{y_1, \cdots, y_l\}$ 是给定的）以及未标记的部分 $X_U = \{x_{l+1}, \cdots, x_{l+u}\}$。

其他形式的监督的可以与提供给系统的约束或**提示**联系起来[1]。例如，提示可以采取的形式是“输出函数必须随着一个输入坐标的增加而增加”，而限制可以是“这两点必须在同一类中”（**必须链接**）或“这两个点不能在同一个类中”（**禁止链接**）。

一个初步的想法起源于 20 世纪 60 年代[96]，即所谓的自学习或**自标记**方法，其中包装算法反复使用一个监督学习方法。该方法一开始是在已标记的实例上进行学习。然后，一些附加的未标记的实例通过使用当前训练得到的系统进行标记，并通过新添加的已标记实例来重复学习。人们可以启发式地尝试给实例添加标记，使该标记具有最高的置信度。虽然有吸引力，但所述包装算法的效果取决于所选的监督学习方法，并且不清楚自标记何时有效。

Vapnik 介绍了一个与 SSL 相关的上下文，作为**直推学习**。我们希望通过归纳学习得到的预测函数适用于任意的输入，但是直推学习的目标只在于预测一套固定的测试点，而这需要动用所有可用信息。通常，直推学习是基于**数据的标记图表示**，已标记的节点是已分类的训练实例，边表示相似性/相异性的关系或约束。执行对标记的组合优化可以使全局一致性度量最大化。

一般情况下，如果关于密度 $p(x)$ 的无监督信息在推导 $p(y|x)$ 时有用，那么 SSL 看起来就很有希望。类比于监督学习中的平滑假设，**半监督的平滑假设**是说，如果高密度区域中的两个输入点 x_1 和 x_2 彼此接近，相应的输出 y_1 和 y_2 也应接近。根据传递性，如果两个点由高密度区域中的一条路径连接（它们属于相同聚类），那么它们的输出应该接近。如果是在低密度区域中，对输出类似的要求就不那么严格了。

如果我们将无监督学习等同于聚类，**聚类假设**是说，如果两个点都在同一个聚类中，那么它们很可能属于同一类。这种情况下，使用未标记的点有助于以更高的准确率来找到聚类间的边界，进而提升满足上述假设的整体分类。

一个等价的阐释是**低密度分离假设**：不同类别之间的边界应该处于低密度区域，并且应该不会分开单个聚类，如图 20-1 所示。

上述假设与国际机场的类比关系非常密切：如果一个人想区分不同的语言，他最好不要分开聚在一起的人，而要在空白区域绘制边界。

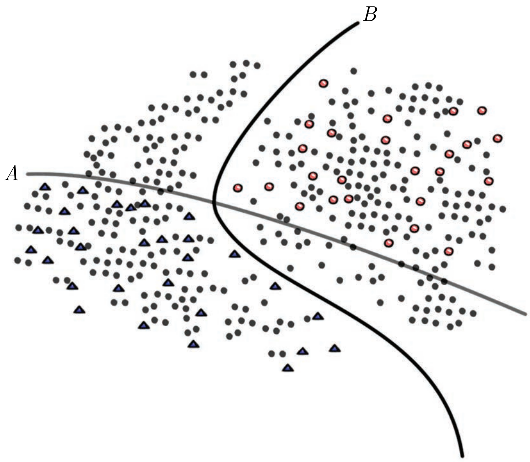


图 20-1　低密度分离假设。尽管界限 A 能完全区分两类，但是界限 B 更好，因为它穿过了一个低密度区域。未标记的数据会产生一个更好的分类器

还有一种不同的范例，假设数据近似地处于一个**低维流形**上，如图 20-2 所示。流形（manifold）是一种数学空间，它在足够小的标度上，类似于一个特定维数的欧几里得空间，这一维数称为流形的维数。例如，一条线和一个圆是一维流形，一个平面和一个球面（球的表面）是二维流形。更正式地说，一个 n 维流形的每个点有一个邻域同胚于 n 维空间 R^n 上的一个开子集。先确定一个流形可以避免维度诅咒（非常高维的输入数据），大多数数据都在这个流形上。然后流形上的**测地线距离**给出一个合适的度量，并且在这个低维流形上考虑标准的平滑性假设。如果有更多的可用数据，人们就能更好地识别应用于监督学习中的相关流形和相应的度量（例如最近邻分类器，所说的邻近度是通过流形上的测地线距离给出的）。

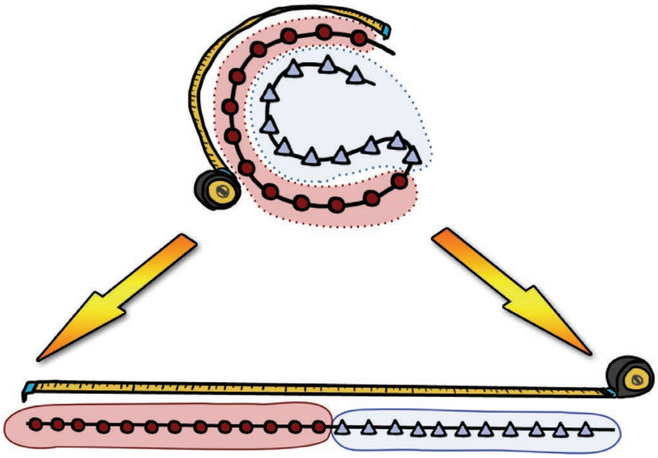


图 20-2　测地线距离可以帮助我们区分在一个流形上的两类

20.1.1 低密度区域中的分离

一些 SSL 的技术的基础是激励类之间（决策边界）的分离通过低密度区域，并远离大多数数据实例。

一个直接的算法可通过采用类似 SVM 的边缘最大化算法得到，无论是已标记还是未标记的实例，它都会最大化边界，这就是所谓的**直推 SVM**（TSVM）。被最小化的函数需要加上如下一项：

$$\lambda_2 \sum_{\text{未标记数据 } i} (1 - |f(x_i)|) \tag{20.1}$$

其中 $f(x_i)$ 是分类函数，其值大于 1 时属于一类，小于 -1 时属于另一类。当 $f(x_i) = 0$ 时，该函数引入的惩罚是 λ_2，并且当 $f(x_i)$ 变为 1 时，或者在另一个方向上，当 $f(x_i)$ 变为 -1 时，它线性地变为 0（惩罚是以 0 为中心的三角形式）。换句话说，如果未标记的数据点落在“灰色”边界区域里，即 $|f(x_i)| \leqslant 1$，就会导致惩罚：无标记的数据往往会**引导线性边界线远离密集区域**。对应的问题是非凸的，因此必须采用健壮的启发式优化方案，例如确定性退火（deterministic annealing）策略，它从一个简单的问题开始，并逐步将其转化成 TSVM 优化函数[99]，或者参考文献[35]中的延续方法，它按照类似的范式：首先优化该函数的“简化”版本，然后逐渐引入越来越精细的细节。

20.1.2 基于图的算法

基于图的方法依赖于将问题表示成图，其中节点对应于实例，边以两个节点 i 和 j 的成对相似性 w_{ij} 为标记。像往常一样，我们可以从相似性的角度，也可以从相异性/距离的角度。

两点沿流形的测地距离的近似值，可以通过从初始成对距离导出点对之间的**最短路径距离**得到。

接下来引入矩阵 $\boldsymbol{W}$ 表示相似性，若边存在，则 $\boldsymbol{W}_{ij} = w_{ij}$，否则为零，且存在对角度矩阵 $\boldsymbol{D}$ 使得 $\boldsymbol{D}_{ii} = \sum_j w_{ij}$。

激励**轻边的平滑性**（连接的节点相似则平滑）的基本方法，与定义和使用**图的拉普拉斯**算子相关。归一化的 $\mathcal{L}$ 和非归一化的组合图拉普拉斯算子 $\boldsymbol{L}$ 定义为：

$$\mathcal{L} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2} \tag{20.2}$$

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} \tag{20.3}$$

图的拉普拉斯算子可以追溯到更传统的拉普拉斯算子（表示为 ∇^2），用于连续函数 $f(x_1, \cdots, x_n)$：

$$\nabla^2 \phi = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial {x_i}^2} \tag{20.4}$$

事实上，点阵的拉普拉斯矩阵，当应用于顶点上 f 的值时，对应于点的规则网络上的连续算子的有限差分近似。图的拉普拉斯矩阵可以看作点阵定义的一般化。

函数 f 在点 $\boldsymbol{x}$ 上的拉普拉斯算子 $\nabla^2 f(\boldsymbol{x})$ 等于一个与维度相关的常数，是函数 f 在以 $\boldsymbol{x}$ 为中心的球面上的平均值随着半径的增加而偏离 $f(\boldsymbol{x})$ 的速率。0 表示这个球面上的平均值等于中心的函数值。

拉普拉斯算子出现在物理学中的动机是，$\nabla^2 f = 0$ 的在一个区域 U 内的解是使狄利克雷能量泛函（Dirichlet energy functional）稳定的函数：

$$E(f) = \frac{1}{2}\int_U \|\nabla f\|^2 \mathrm{d}x \tag{20.5}$$

平滑行为是显然的：人们旨在找出局部最优配置，来最大限度地减少梯度模的均方。从优化的角度我们再次澄清了意义。

为点阵上的上述 $\nabla f = 0$ 方程求解的迭代方法，是对于点阵中的每个点，反复地以它的邻居的加权平均值来代替它的值。

图上也可以有类似的平滑行为，我们的目标是得到的图上的值的分布，以使得**在节点处的值等于其相邻值的加权平均**。

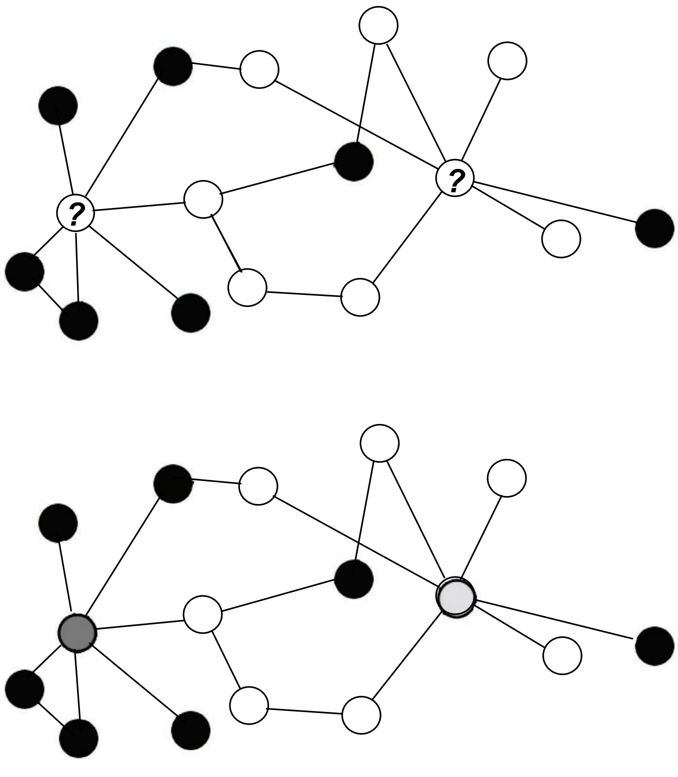


图 20-3 通过图中相邻点的平均值计算未知点的值

参考文献[120] 中提出了一种采用高斯场和谐波函数的半监督学习。高斯场的分类算法可

以看作最近邻方法的一种形式，其中最接近的已标记实例是由一个图中的随机游走来计算。这一方法的公式涉及电子网络和谱图理论。该问题表示为图，其中一些节点标记为 $y \in 0,1$（为简单起见，采用二进制标记）。加权边表示相似性：w_{ij} 会很大，如果实例类似。例如 $w_{ij} = \exp\{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2_{\boldsymbol{A}}\}$ 就是一个合适的度量。该策略先计算出所有节点的一个“平滑”实值函数 f，然后基于 f 值赋予标记。相似点具有相似的值，这一“平滑”的愿望表示为最小化二次能量函数：

$$E(f) = \frac{1}{2}\sum_{i,j} w_{ij}(f(i) - f(j))^2 \tag{20.6}$$

最小能量函数是**和谐的**：对于未标记的点，它满足 $Lf = 0$；对于已标记的点，则等于其标记的值。L 是之前定义的图上的拉普拉斯算子，而和谐的特性意味着 f 在未标记的点上的值等于相邻点 f 值的加权平均：

$$f(j) = \frac{\sum_i w_{ij} f(i)}{\sum_i w_{ij}} \tag{20.7}$$

用矩阵表示法：$f = \boldsymbol{P}f$，其中 $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{W}$。这与相似关系中的平滑的直观概念是一致的。图 20-3 展示了基于图的平滑操作。

一个简单的规则是，若 $f(i) > 1/2$，则将节点 i 标记为 1，否则为 0。

与**随机游走**的联系如下：想象一个行走的人从一个未标记的节点 i 开始，并以概率 $\boldsymbol{P}_{ij}$ 移动到邻近点 j。遇到第一个已标记点时，游走停止。于是，$f(i)$ 是停止在一个标记为 1 的节点的概率。

电子网络解释如下：标记为 1 的节点连接到正电压源，标记为 0 的节点接地。边是电导为 w_{ij} 的电阻。然后 f 是未标记的节点上产生的电压，使得能量耗散最小化。把类的先验知识（这两个类的理想的比例）通过修改节点标记的阈值加入。参考文献[120]中描述了从已标记和未标记的数据中学习权重矩阵 $\boldsymbol{W}$ 的可能方法。

20.1.3 学习度量

一些半监督算法按两个步骤进行：首先通过对所有数据（忽略标记的存在）的无监督步骤来确定**一个新的度量或表示法**，然后使用新确定的度量或者表示法来执行纯监督学习阶段。

这两个步骤实际上是在实现半监督平滑假设，通过确保新的度量或表示满足在密度高的区域距离小的条件。

注意，某些基于图的方法与这种处理方式密切相关：根据数据进行图的构建，可以看作一个无监督的表示法变化。

20.1.4 集成约束和度量学习

许多情况下，当处理有一个以上变量的优化问题时，可以采用序列方法。它首先针对第一个变量求最小值，然后针对第二个变量（保持第一个变量不变），等等。相比于同时考虑所

有这些变量，这种序列方法给出的结果一般都需要改进。这一点很清楚，因为同时考虑所有变量使得在输入空间移动的自由度增加：第一种情况下，只能沿着坐标轴移动；第二种情况下，可以自由地在输入空间内移动，寻找局部最优点。

这同样适用于 SSL。例如，参考文献[24]中的工作展示了如何结合约束和度量学习来进行半监督聚类。

基于约束的聚类方法从点对之间的必须链接或禁止链接的约束（即两个点属于或不属于同一个聚类的要求）开始，在需要最小化的目标函数中加入违反约束的惩罚。顺便说一下，约束可以来自标记，也可以来自其他信息源。例如，欧几里得 K 均值算法将点分成 k 组，使得函数

$$\sum_i \|\boldsymbol{x}_i - \boldsymbol{\mu}_{l_i}\|^2$$

局部极小化。其中，向量 $\boldsymbol{\mu}_{l_i}$ 是点 $\boldsymbol{x}_i$ 所属聚类的中心点，即到 $\boldsymbol{x}_i$ 最近的那个。

如果知道必须链接点对 $\mathcal{M}$ 和禁止链接点对 $\mathcal{C}$ 这两个集合，就可以激励一个满足约束的中心点的摆放，只要违反了 $\mathcal{M}$ 中的某个约束，就加上惩罚 w_{ij}；同理，只要违反了 $\mathcal{C}$ 中的某个约束，就加上惩罚 $\overline{w}_{ij}$，这样就得到以下需要最小化的函数（**“成对约束K均值”**）：

$$E_{\text{pckmeans}} = \sum_i \|\boldsymbol{x}_i - \mu_{l_i}\|^2 + \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\mathcal{M}\text{且}l_i\neq l_j} w_{ij} + \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\mathcal{C}\text{且}l_i=l_j} \overline{w}_{ij} \tag{20.8}$$

成对约束也可用于**度量学习**。如果以对称正定矩阵 $\boldsymbol{A}$ 参数化度量如下:

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\boldsymbol{A}} = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{x}_i - \boldsymbol{x}_j)}$$

问题就变成了确定矩阵系数的相应值。如果该矩阵是对角矩阵，问题就变成了确定不同属性的权重。

约束代表用户对相似性的看法：通过最小化必须链接的实例间的距离，同时最大化禁止链接的实例间的距离，相似性可以用来改变度量，以反映这种观点。度量被修改之后，人们可以使用像 K 均值这样的传统聚类算法。

要求在整个空间中使用单个度量也许是不恰当的，对于每个 K 均值聚类 h，可以用不同的度量 $\boldsymbol{A}_h$。参考文献[24]中的 MPCK-MEANS 算法采用期望最大化方法，见 15.3 节，也改变了 E 步骤（期望步骤）的聚类分配，以及 M 步骤（最大化步骤）的中心点估计和度量学习。

约束在聚类初始化以及把数据点分配给聚类时使用。每次迭代期间，基于当前聚类分配和约束违反的情况，距离度量通过重新估算 $\boldsymbol{A}_h$ 进行调整。参考文献[75]是一篇有趣的讨论文本文档的度量学习的论文。关于半监督学习的更多细节可以在参考文献[36]和参考文献[119]中找到。



梗　　概

许多情况下，已标记的实例稀缺且难得，未标记的实例却很多，它们平时沉睡在商业数据库里或网页上。

半监督学习方案同时使用可用的已标记实例和未标记实例，以提高整体的分类准确率。

所有实例的分布可以用于激励 ML 分类方案，从而创建类别之间的穿过**低密度区域**的边界（直推 SVM）。

如果问题以**图**来建模（实体和以距离标记的关系），图上的平滑操作可以用来使一些已标记节点的信息传送到相邻节点（图的拉普拉斯算子）。

实例的分布可以用于**学习一个度量**，而度量是继续进行监督学习的关键组成部分。

一个外星人到了地球上，在征服我们之前，可以先结合网页上不计其数的信息，加上从人类笔友（或像雅虎一样的目录）那里获得的一些已标记信息，以密集的方式学习了解人类文明。地球上的企业使用类似的技术来挖掘数据和征服更多的客户。

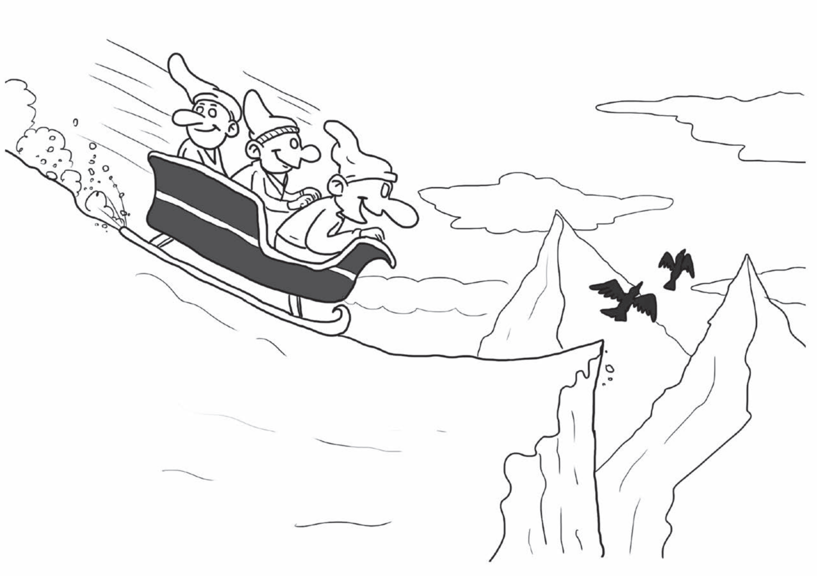
第三部分

优化：力量之源

第 21 章　自动改进的局部方法

我要去走走 —— 在这个世界上;
直到我的脚 —— 不再让我走下去;
是的，我就是要去走走 —— 并将走得更远。

—— 梅西　格雷和祖凯罗　福尔纳恰里



很多问题都可以转换成替一个合适的目标函数寻找最优值的问题，当然是在某些约束的前提下。如果你正准备买房子，你会有预算和一些目标，比如房间数、邻里情况、景致、离工作地点的远近、学校等。如果你正在寻找一个伴侣，你的目标可能会是智慧、美丽、在一起的感受等。如果你正在经营一个公司，在给定人力资源和设备等约束的前提下，你会把目标定在最大限度地提高利润　　你可能会注意到，定义适当的目标函数，这本身就不是一份简单的工作（想想你对于伴侣的偏好函数）。然而，一旦完成了这个关键的前期工作，剩下的重任就是**最小化或最大化这个函数**。**这个函数将自变量映射成输出值**。最大化意味着找到使得输出值最大的输入值。

函数的优化方法是解决大多数问题和进行决策的**力量之源**。这一重要性有些时候很明显，有些时候又没那么明显，无论如何，我们都有合理的动机去理解这些基本思路和工具。这一

主题对于学习和智能优化（LION）方法尤其有意义，因为 LION 结合了数据挖掘、建模和交互式问题解决和优化。虽然使用该技术之前，底层的方法并不是非知道不可的，但掌握这些基础有助于更快和更有效地做出选择。

现在考虑以下相关问题。

- **非线性方程组的问题**，即求解一组非线性方程组（所有函数 f_i 都包含在向量 F 里）：

$$\begin{aligned}&\text{给定 } F:\mathbb{R}^n \longrightarrow \mathbb{R}^n\\&\text{求 } x^*\in\mathbb{R}^n \ \text{ 使得 } F(x^*)=0\in\mathbb{R}^n\end{aligned}$$

  若解 x^* 存在，则它最小化 $\sum_{i=1}^{n}(f_i(x))^2$。这是显而易见的，因为平方和非负，并且当且仅当所有函数值为零时，它等于零。

- **无约束的最小化**：

$$\begin{aligned}&\text{给定 } f:\mathbb{R}^n \longrightarrow \mathbb{R}\\&\text{求 } x^*\in\mathbb{R}^n \ \text{ 使得对于任意 } x\in\mathbb{R}^n \ \text{ 满足 } f(x^*)\leqslant f(x)\end{aligned}$$

  满足上述条件的一个点 x^* 被称为*全局最优*，根据定义，它是该问题最有可能的解：不存在更好的其他解。

本章中会介绍带有连续变量（实数）的优化函数的一些基本和传统的方法，并演示它们的收敛性。

如果你不喜欢数学，你可能想跳过这一部分而继续下面的内容。下面的章节更务实，会谈论离散问题的*局部搜索和反馈搜索优化*（第 22 章的 RSO）、连续和合作优化的 RSO（第 23 章的 CoRSO）以及多目标优化的 RSO（第 24 章的 MORSO）。

21.1 优化和学习

学习和优化技术之间有很强的联系。

一方面是**把优化用于学习**，从一类模型中选择与数据最为一致的那个（最能解释观察到的数据那个）。例如，常用于曲线拟合和监督学习的“差的平方和”。当然，学习的最终目的是*泛化*，但这仅意味着需要被最小化的函数将包含更多部分，考虑到模型的*复杂度*，因此简单的模型比复杂的模型更受青睐。

另一方面是**把学习用于优化**，某些形式的*学习*也用于高效优化算法。“学习”方式的基本例子，虽然发明者没有使用这个术语，但可以在连续优化的标准技术中找到。这些技术中构建了局部模型（通过使用函数及其导数的局部信息得到），它的合法性被限制在当前点周围（*模型信赖域法*）。无论模型还是*信赖域*（trust-region）通常都通过寻找局部极小来调整。

虽然这些技术传统上与连续优化相关联，但同样的原理——**通过优化学习一个局部模型**（或者根据实例和局部特性调整的参数）——可以用于不同的离散（组合）优化，参见第 22 章所描述的局部搜索和反馈搜索优化（RSO）技术 [8]。

很多方法的主旨（leitmotiv）是**从一个试探性的解开始，通过一系列步骤进行修改**，从而找到最终解。在每一个步骤里，我们为被优化的**函数**构建一个**局部模型**，并将这个模型用于局部移动，对试探性的解进行微调。因此，该方法缺乏大局观，不能保证收敛到全局最优。然而在实践中，虽然有局部极小值的存在（会使得局部搜索器卡住），但瑕不掩瑜，梯度下降、局部搜索以及其他相关技术仍然可能是最简单和最成功的解决问题的方法。

现在来看，在连续函数中，一个从给定实例的优化中习得的灵活（带参数）局部模型的原理如何起作用。下面各节目的是让大家见识连续优化的最基本和最成功的范例，重点在于直觉而非数学细节。数学细节可在参考文献 [42] 中找到。

此时需要区别将被最小化的函数的**可导性**。在现实世界中，函数大多是不可导的，事实上许多例子中输入与输出的对应关系是非连续的，或者输入就是离散的（比如整数）。如果尝试询问一个商人其利润的导数，以作为重要业务决策的函数，估计你将得不到任何答案!

如果你恰好要处理一个实数函数 $f(x)$，并且它是连续可微的，那么可以使用一系列标准方法。下面首先特别总结了一维的最优化方法（21.2 节），然后回顾了多维空间下最优化模型的求解技巧（21.3 节），最后介绍了使用求解模型技术来最优化多变量非线性函数（21.4 节）。

如果你不巧要处理一个不能求导的函数，那么你可以跳过前几节，只使用基于函数计算的方法，就像 21.5 节或者第 23 章和第 24 章中介绍的方法。

21.2 基于导数技术的一维情况

一维的情况可能更加直观，因此我们先考虑一些一元函数的经典结果。一个历史悠久并继续影响当今的找到可微函数 $f(x)$ 零点的基本方法，是从一个离目标值足够近的点开始，进行如下两个步骤的迭代：

(1) 找到一个**局部可解模型**；

(2) 解这个模型。

围绕当前点 x_c 的局部模型可以由前三项的**泰勒级数近似**（Taylor series approximation）得到：

$$f(x) = f(x_c) + f'(x_c)(x - x_c) + \frac{f''(x_c)(x - x_c)^2}{2!} + \cdots$$

或者通过牛顿定理：

$$f(x) = f(x_c) + \int_{x_c}^{x} f'(z)\mathrm{d}z \approx f(x_c) + f'(x_c)(x - x_c)$$

因此，围绕当前估计 x_c 的一个**局部模型**（实际上是**仿射模型**）是：

$$M_c(x) = f(x_c) + f'(x_c)(x - x_c)$$

通过寻找这一模型的根，人们能够得到当前估计对应的下一个值 x_+ 的式子（从 x_c 到 x_+ 的局部步骤），如图 21-1 所示：

$$x_+ = x_c - \frac{f(x_c)}{f'(x_c)}$$

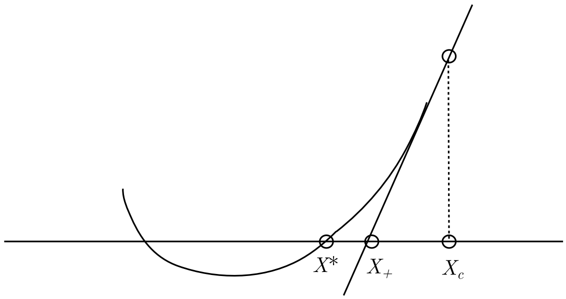


图 21-1　牛顿法的局部模型

如果函数是线性的，收敛就可以一步完成。如果函数不是线性的，让我们来学习牛顿法的*局部收敛*（local convergence）性质：证明，如果从一个离根足够近的点 x_c 开始，那么总会收敛到这个根。这个证明基于对模型的*线性性缺失限制*（bounding the lack of linearity），以及每一步都会使当前点距目标根的距离减少（contracted）。

线性性的缺失，或者使用该模型造成的误差是：

$$f(x) - M_c(x) = \int_{x_c}^{x} [f'(z) - f'(x_c)]\mathrm{d}z$$

现在需要限制函数值的变化，令其正比于其输入的差异。

定义 1（Lipschitz continuity，李普希茨连续性）　*一个函数 g 是以常数 γ 在集合 X 上李普希茨连续的（ $g \in Lip_\gamma(X)$ ），如果对所有 $x, y \in X$，有：*

$$|g(x) - g(y)| \leqslant \gamma |x - y|$$

引理 1　*假设 $f' \in Lip_\gamma(D)$ 定义在开区间 D 上。那么对于任意的 $x, y \in X$：*

$$|f(y) - f(x) - f'(x)(y - x)| \leqslant \gamma \frac{(x - y)^2}{2}$$

证明

$$|f(y) - f(x) - f'(x)(y - x)| = \int_0^1 [f'(x + t(y - x)) - f'(x)](y - x)\mathrm{d}t$$

根据三角不等式和李普希茨连续性：

$$\leqslant |y-x|\int_0^1 \gamma|t(y-x)|\mathrm{d}t = \gamma|y-x|^2/2$$

现在可以证明**一维牛顿法的收敛定理**。图 21-2 可以辅助理解这个证明。

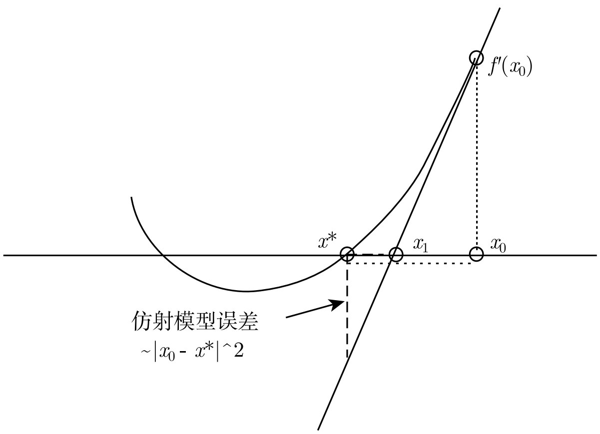


图 21-2 当起始点 x_0 离 x^* 很近时，可以保证收敛

定理 1 在开区间 D 上，令 $f: D \longrightarrow \mathbb{R}$，$f' \in Lip_\gamma(D)$（李普希茨连续性），在 D 上，$|f'(x)| \geqslant \rho$（导数有一个离开零点的界）。

如果 $f(x)=0$ 有一个解 $x^* \in D$ 并且起始点 x_0 足够近，那么这个解可以用牛顿法找到。

如果存在 $\eta > 0$ 使得 $|x_0 - x^*| < \eta$，那么序列

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

存在并且收敛至 x^*。另外，

$$|x_{k+1} - x^*| \leqslant \frac{\gamma}{2\rho}|x_k - x^*|^2$$

证明 找到一个起始的球（ball）满足：

$$|x_{k+1} - x^*| \leqslant \tau|x_k - x^*| \text{ ，} \tau \in (0,1)$$

这也意味着点将留在这个球内：

$$x_1 - x^* = x_0 - x^* - \frac{f(x_0)}{f'(x_0)} = x_0 - x^* - \frac{f(x_0)-f(x^*)}{f'(x_0)}$$

$$= \frac{1}{f'(x_0)}[f(x^*) - f(x_0) - f'(x_0)(x^* - x_0)] = \frac{1}{f'(x_0)}[f(x^*) - M_0(x^*)]$$

其中基于 x_0 的仿射模型在 x^* 上的误差被限制在 $\frac{\gamma}{2}|x_0-x^*|^2$。因此，

$$|x_1-x^*| \leqslant \frac{\gamma}{2|f'(x_0)|}|x_0-x^*|^2 \leqslant \frac{\gamma}{2\rho}|x_0-x^*|^2$$

如果 $\frac{\gamma}{2\rho}|x_0-x^*|<1$，或者

$$|x_0-x^*| \leqslant \frac{2\rho}{\gamma}\tau$$

距离就会缩小，并且如果从一个半径为

$$\eta = \tau\frac{2\rho}{\gamma} \quad （可能缩小以适合区间 D）$$

的球开始，距离也会缩小。

上面的定理确保了一个快速（二次）收敛，前提是从一个距离目标根足够近的位置开始。如果已经得到了足够好的近似解，情况会是如此；然而，一般来说开始的位置是非常远的，并且无法保证这一起始点经过这些步骤之后会最终得到目标解。

这一问题是**全局收敛**（global convergence）。在现实中，由于缺乏强有力的保证，就将很多方法**混合起来**，当牛顿法可行时就用牛顿法，否则就回到某个相对较慢但是安全的全局方法，比如**二分法**（bisection），如图 21-3 所示。

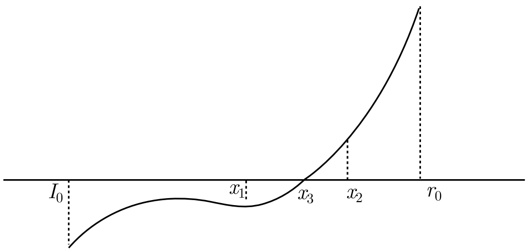


图 21-3　二分法示意图

二分法通过将一个起始区间（从 l_0 到 r_0）细分为两部分来寻找一个连续函数的根，过程中观察 f 在中间点的值，并且只继续搜寻左或右子区间（当然要确保所选的子区间包含根）。二分法简单而有效，并且它在一个对数步骤内收敛。然而，这一简单的方法很难扩展到一维以上的情况。

图 21-4 描述了**回溯**（backtracking）的思想：如果牛顿法的步骤走得太远，超过了根的区域，就可以调转方向回到离根更近的位置。我们从点 x_N 朝着起始点 x_c 移动，直到找到 x_+ 使得 $|f(x_+)|<|f(x_c)|$。

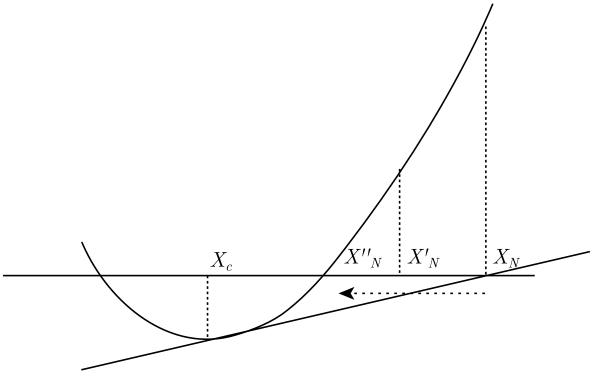


图 21-4　回溯：牛顿法的每一步都有方向

混合方法的一个通用模式是结合全局收敛性和局部快速收敛性，如图 21-5 所示。我们可以首先尝试牛顿法的步骤，但要确保这一迭代减少距离解的某种度量。

```
function hybrid_quasi_newton (f : ℝ → ℝ, x_0)
    while not 结束
        构造 x_k 附近的局部模型f，并找到它的一个解 x_N
        if x_{k+1} 可接受 then 移动
        else 用一个安全的全局策略选择 x_{k+1}
```

图 21-5　混合拟牛顿法示意图

21.2.1　导数可以由割线近似

如果导数无法计算或很难计算，那么可以用通过这两个点的割线（secant）近似（以一个有限差分近似）。这一**割线法**使用前一迭代 x_- 如下：

$$a_c = \frac{f(x_c) - f(x_-)}{x_c - x_-}$$

对此，有一个收敛定理成立，虽然收敛速度现在慢一些（线性）。

定理 2　对于开区间 D，令 $f : D \longrightarrow \mathbb{R}$，$f' \in Lip_\gamma(D)$（李普希茨连续性），在 D 上，$|f'(x)| \geqslant \rho$（导数有一个离开零点的界）。

如果方程 $f(x) = 0$ 有一个解 $x^* \in D$，那么存在正常数 η, η' 使得，如果 $0 < |h_k| \leqslant \eta'$ 且 $|x_0 - x^*| < \eta$，则序列

$$x_{k+1} = x_k - \frac{f(x_k)}{a_k}\ ,\ a_k = \frac{f(x_k + h_k) - f(x_k)}{h_k}$$

q- 线性性收敛至 x^*。

我们认识到，基于导数的方法可以用作开发无导数的方法的起始点。这些近似会牺牲一些效率，但收敛还是可以保证的。

21.2.2 一维最小化

直到现在，我们都在讨论求根，使得函数 f 的值等于零。为了最小化一个可导的函数，我们从这个必要条件开始[①]：最小值必须满足 $f'(x^*) = 0$。所有的工作就是找到导函数的一个根，并且我们已经知道了如何解决这一问题！可以用混合牛顿法，加上要求 $f(x_k)$ 递减。用一阶导数 f' 代替原始函数 f，可以得到：

$$x_+ = x_c - \frac{f'(x_c)}{f''(x_c)}$$

注意，f' 的仿射模型隐含着 f 的一个围绕 x_c 的**二次模型**：

$$m_c(x) = f(x_c) + f'(x_c)(x - x_c) + \frac{1}{2}f''(x_c)(x - x_c)^2$$

这一迭代将会局部收敛，并且如果 $f''(x^*) \neq 0$ 且 f'' 在 x^* 附近满足李普希茨条件，那么它会 Q–二次收敛到 x^*。如果有必要，可以一直回溯到 $f(x_+) < f(x_c)$。

21.3 求解高维模型（二次正定型）

采用局部二次模型进行优化之前，让我们把动机变得更强，确保这些局部模型其实是可以解决的。现在考虑多元函数。解决局部二次模型相当于解决一个二次型。图 21-6 展示了二次正定型（quadratic positive-definite form）的一个例子。

现在，**牛顿法**要求该模型的**梯度**等于零。给定一步 s，对应的二次模型是：

$$Q(s) = \sum_{i=1}^{n} g_i s_i + \sum_{i=1}^{n}\sum_{j=1}^{n} H_{ij} s_i s_j \equiv g^{\mathrm{T}} s + \frac{1}{2} s^{\mathrm{T}} H s$$

求得梯度以后，我们要求：

$$\nabla Q(s) = 0 = g + Hs \tag{21.1}$$

$$Hs^N = -g \quad \text{（牛顿方程）} \tag{21.2}$$

这一线性方程组可以通过对一个复杂度为 $O(n^3)$ 的矩阵求逆来求解。[②]

---

① 充分条件是 $f''(x^*) > 0$，比如使用带余项的泰勒展开式：

$$f(x) - f(x^*) = f'(x^*)(x - x^*) + \frac{1}{2}f''(\bar{x})(x - x^*)^2$$

② 实际上，如果使用更细粒度的技术，矩阵求逆的复杂度可以为 $O(n^{\log_2 7})$，甚至更小。但由于复杂度和数值计算的问题，实际上通常不会使用这种方法。

由于计算机的计算精度有限，需要处理**数值稳定性**（numerical stability）问题：一些技术会积累误差，这一现象很危险，很可能使得得到的数值解与精确的数学解（只有当计算机可以用无限精度来表示实数时才可计算）非常的不同。

病态（ill-conditioning）是用于度量数值解对数据变化的敏感程度（由于有限精度的计算）的术语。图 21-7 展示了一个二维的例子（两个相似的方程对应平面上几乎平行的直线）。

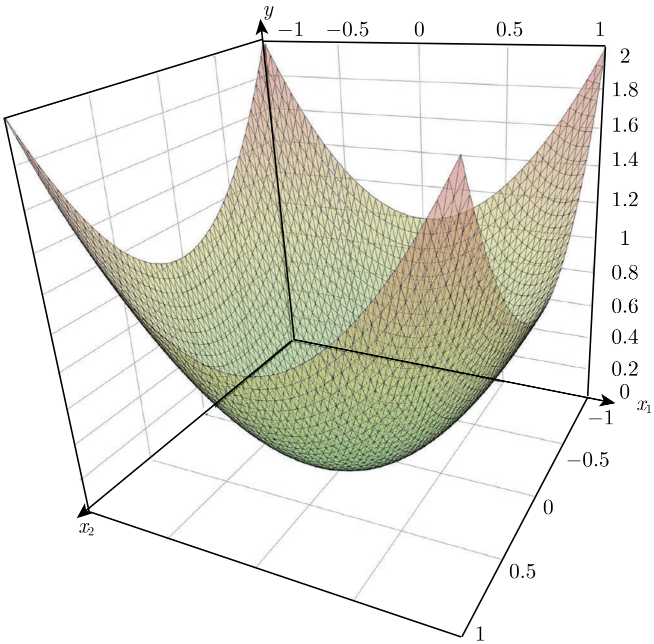


图 21-6　含有两个变量的二次正定型 f

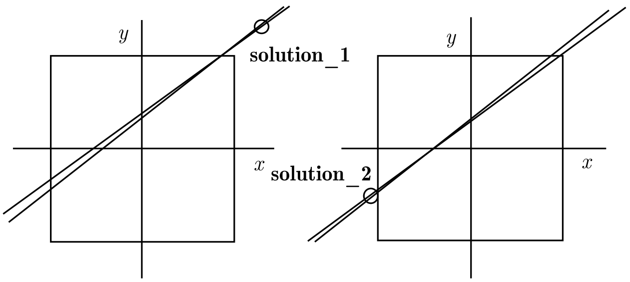


图 21-7　病态：数值解对于数据变化非常敏感，该图中两个线性方程十分相似，稍微改变线的方向将会极大地影响数值解

从细节上来说，人们给矩阵 $\boldsymbol{H}$ 定义了**条件数** $\kappa(\boldsymbol{H}) = ||\boldsymbol{H}||\,||\boldsymbol{H}^{-1}||$，其中 $||*||$ 是由向量

范数 $\|\boldsymbol{H}\| = \max_x(\|\boldsymbol{H}x\|/\|x\|)$ 导出的矩阵算子范数。条件数是 $\boldsymbol{H}$ 导出的最大与最小拉伸的比值。它度量除去其他影响之外的**以有限精度运算时的线性系统解的敏感性**。如果一个线性系统 $\boldsymbol{H}x = b$ 以如下的方式被扰动，添加一个正比于 ϵ 的误差：

$$(\boldsymbol{H} + \epsilon F)s(\epsilon) = g + \epsilon f \tag{21.3}$$

那么解中间的相对误差可以有如下的界：

$$\frac{\|s(\epsilon) - s\|}{\|s\|} \leqslant \kappa(\boldsymbol{H})\,(\frac{\|\epsilon F\|}{\|\boldsymbol{H}\|} + \frac{\|\epsilon f\|}{\|g\|}) + O(\epsilon^2)$$

关于对称正定矩阵，**楚列斯基分解**（Cholesky factorization）可以帮助我们找到一个十分稳定的三角分解。将 $\boldsymbol{H}$（对称且正定）写为：

$$\boldsymbol{H} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$$

其中 $\boldsymbol{L}$ 是单位下三角矩阵，$\boldsymbol{D}$ 是正元素的对角矩阵（$\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$ 分解）。

因为对角元素是正的：

$$\boldsymbol{H} = \boldsymbol{L}\boldsymbol{D}^{1/2}\boldsymbol{D}^{1/2}\boldsymbol{L}^{\mathrm{T}} = \bar{\boldsymbol{L}}\bar{\boldsymbol{L}}^{\mathrm{T}} = \boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}$$

其中 $\boldsymbol{R}$ 是一般性的上三角，所以楚列斯基因子也是 $\boldsymbol{H}$ 的“平方根”，是通常的平方根在矩阵上的一次扩展。

$\boldsymbol{R}$ 可以通过矩阵中的元素相等来计算：

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} r_{11} & & & \\ r_{21} & r_{22} & & \\ \vdots & \vdots & \ddots & \\ r_{n1} & r_{n2} & \dots & r_{nn} \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix}.$$

我们为元素 (1,1) 列出等式：

$$a_{11} = r_{11}^2,\ \ r_{11} = \sqrt{a_{11}}$$

接下来为第一行列出等式：

$$a_{12} = r_{11}r_{12},\ \ a_{13} = r_{11}r_{13}, \cdots$$

第一行结束以后，开始第二行，元素如下：

$$a_{22} = r_{12}^2 + r_{22}^2$$

上述过程需要大约 $\frac{1}{6}n^3$ 次乘法和加法，以及 n 次求平方根（如果用 $\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}$ 就可以避免求平方根）。R 中的元素不会急剧增长，因为下式总是成立：

$$a_{kk} = r_{1k}^2 + r_{2k}^2 + \cdots + r_{kk}^2$$

现在，原式变成了：

$$\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}s = g \tag{21.4}$$

这个式子可以用向后替代法求解（不断求解变量，并将其带入剩下的方程中）。以如下的方式剥离因子：

$$\boldsymbol{R}^{\mathrm{T}}s_1 = -g \quad \text{使用向前替代} \tag{21.5}$$

$$\boldsymbol{R}s = s_1 \quad \text{使用向后替代} \tag{21.6}$$

解方程的成本是 $O(n^2)$，因此显性成本是在分解的步骤中。

21.3.1　梯度与最速下降法

在很多情况下，如果线性系统非常大，通过矩阵求逆来找到二次模型的最小值既不是最高效也不是最健壮的方式。这种情况在机器学习中十分频繁。另外，很多时候二次偏导矩阵 $\boldsymbol{H}$ 难以计算，或者计算成本太高。

针对上述情况，**梯度下降法**（gradient descent）提供了一个简单可行的策略，它朝着局部最优逐渐改进一个初始解。

如果梯度不为零，并且朝着负梯度的方向移动：

$$x_+ = x_c - \epsilon \nabla f$$

考虑 f 的泰勒展开式 (4.4)，那么存在一个足够小的 ϵ，使得函数值下降，即 $f(x_+) < f(x_c)$。尽管比较粗糙，并且需要小心谨慎地选择一个小的 ϵ 值，但是上述技术在很多领域中都发挥着作用（例如在第 9 章中所介绍的训练神经网络的流行算法，即误差反向传播方法）。

最速下降法（steepest descent）的解释非常自然和直观。在某个表面上的一滴水，会根据局部的梯度流向局部最低处，至少大概情况是这样的。滑雪的人，就像图 21-8 中那样，都对最速下降法有着深刻的体会，并且必须让滑雪板与最速下降的梯度垂直来停住。最速下降的离散版本将在第 22 章以**局部搜索**的形式进行讨论。其思路是搜索过程首先在邻域中对函数值进行抽样，然后再决定走哪一步。在这些过程中，**没有用到全局视野，只有局部信息**。

除了作为下降方向，众所周知的是，负梯度 $-g$ 也是最快的下降方向。有一个表面上似乎可靠的方法，是通过沿着以下梯度进行一维的最小化：

$$\min_t Q(x_c - gt)$$



图 21-8 两位梯度下降法的专家在意大利特伦托附近的雪山上

可惜，这种直觉是错误的：很多情况下，把力气花在沿着梯度进行最小化上，不是解决最小化问题的最优方法。

问题在于，当矩阵是**病态的**时，梯度方向将不会指向最优解，反而会越来越指向一个垂直的方向！二维空间里的病态能被可视化：等高线被朝着某个方向拉伸，见图 21-9。当沿着梯度前进时，轨迹会变得**蜿蜒曲折**，使得到达最小值的时间增加。

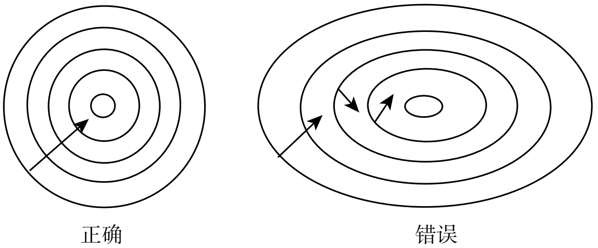


图 21-9 当搜索最小值的时候，梯度有时并不是合适的方向

可以证明，如果用最速下降法来最小化一个二次函数 $Q(s) = g^{\mathrm{T}}s + \frac{1}{2}s^{\mathrm{T}}\boldsymbol{H}s$（$\boldsymbol{H}$ 是对称且正定的），收敛可能会非常缓慢。用条件数 κ 来具体解释，当 κ 增加时，当前值与最优值之间的距离在每一次迭代时都要乘以一个接近 1 的数：

$$
\begin{aligned}
|Q(s_{k+1}) - Q(s_*)| &\approx \left(\frac{\eta_{\max} - \eta_{\min}}{\eta_{\max} + \eta_{\min}}\right)^2 |Q(s_k) - Q(s_*)| \\
&\approx \left(\frac{\kappa - 1}{\kappa + 1}\right)^2 |Q(s_k) - Q(s_*)|
\end{aligned}
$$

如果允许我们打一个牵强的比方，上述情况对生活有些指导意义：如果总是按照贪心法

的方式，以局部最优方向上的最小化为目标，可能会失去更多“全局”的机会。

21.3.2　共轭梯度法

非干扰方向这一概念是将共轭梯度法用于最小化的动机。两个方向是关于矩阵 $\boldsymbol{H}$ 互相共轭（mutually conjugate）的，若有

$$p_i^{\mathrm{T}}\boldsymbol{H}p_j = 0，i \neq j \tag{21.7}$$

沿着方向 p_i 最小化，之后这个极小点处的梯度将会垂直于 p_i。如果接下来的最小化沿着方向 p_{i+1}，那么沿着这一梯度方向的变化是 $g_{i+1} - g_i = \alpha\boldsymbol{H}p_{i+1}$（对于某个常数 α）。这个矩阵 $\boldsymbol{H}$ 其实是黑塞（Hessian）矩阵，即包含二次导数的矩阵，并且在二次的情况下，这一模型恰好是原来的函数。现在，如果式 (21.7) 成立，这一变化将*垂直于*前一个方向（$p_i^{\mathrm{T}}(g_{i+1} - g_i) = 0$），因此，*这一新点的梯度保持与* p_i *垂直*，并且前一步最小化是可接受的。虽然对于二次函数，这种共轭梯度法可以保证在至多 $n+1$ 个函数和梯度运算之后收敛到最小值（至少对于无限精度的计算）；然而对于一般的函数，这些步骤需要迭代直到得出最小值的一个合适的逼近。

引入向量 $y_k = g_{k+1} - g_k$。首先搜索的方向 p_1 由负梯度 $-g_1$ 给定。接着，序列 x_k，作为最小化的逼近，定义为：

$$x_{k+1} = x_k + \alpha_k p_k \tag{21.8}$$

$$p_{k+1} = -g_{k+1} + \beta_k p_k \tag{21.9}$$

其中 g_k 是梯度，α_k 被选来沿着搜索方向 p_k 最小化 E，β_k 由下式给定：

$$\beta_k = \frac{y_k^{\mathrm{T}} g_{k+1}}{g_k^{\mathrm{T}} g_k} \qquad \text{（Polak-Ribiere 方法）} \tag{21.10}$$

或者：

$$\beta_k = \frac{g_{k+1}^{\mathrm{T}} g_{k+1}}{g_k^{\mathrm{T}} g_k} \qquad \text{（Fletcher-Reeves 方法）} \tag{21.11}$$

对于一个二次函数，这两种方法是一样的 [97]。上述形式的一个主要难点在于，对于一个一般的函数，得到的方向不一定是下降的方向，并且可能导致数值不稳定。

使用*动量*（momentum）项以避免在*反向传播*（back-propagation）[92] 中的振荡被当作共轭梯度的近似形式。

21.4　高维中的非线性优化

现在来考虑牛顿法在高维中的收敛性质。这一方法需要求解如下的二次模型：

$$m_c(x_c + p) = f(x_c) + \nabla f(x_c)^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} \nabla^2 f(x_c) p$$

并且迭代，如图 21-10 所示。

function `multi_dimensional_newton` $(f: \mathbb{R}^n \to \mathbb{R},\ x_0 \in \mathbb{R}^n)$

  f 是二次连续可微的

  while not 结束

    解 $\nabla^2 f(\boldsymbol{x}_c) S^N = -\nabla f(x_c)$

    $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + S^N$

图 21-10 高维中的牛顿法

如果初始点接近最优解 x^* 并且 $\nabla^2 f(x^*)$ 是正定的，那么该方法 Q- 二次收敛到 x^*。

假如下列情况发生，会产生一些问题。

- 黑塞矩阵非正定：存在负曲率的方向 $p^{\mathrm{T}}\boldsymbol{H}p < 0$，这意味着当朝着方向 p 走出无穷远时，这个二次局部模型可以假设有任意大的负值。
- 黑塞矩阵是奇异的或病态的，使得矩阵求逆变得不可能，或至少很困难。

上述问题引出了所谓的**修正牛顿法**，将局部模型变得足够正定和非奇异。此外，这些方法处理全局收敛以及不定矩阵 $\boldsymbol{H}$，并且使用迭代来逼近 $\boldsymbol{H}$。这一方法将**一个快速的局部战术与一个健壮的全局战略相结合**，从而保证全局收敛。

21.4.1 通过线性查找的全局收敛

全局收敛是通过沿着确定的方向进行线性搜索而得到的：一开始尝试牛顿法，然后可能再回溯。当然，需要确保这一方向的确是下降方向！幸运的是，如果矩阵 $\boldsymbol{H}$（对称的）是正定的，那么牛顿方向总是下降方向：

$$\frac{\mathrm{d}f}{\mathrm{d}\lambda}(x_c + \lambda s^N) = \nabla f(x_c)^{\mathrm{T}} s^{\mathrm{T}} = -\nabla f(x_c)^{\mathrm{T}} \boldsymbol{H}_c^{-1} \nabla f(x_c) < 0$$

如果黑塞矩阵必须被逼近，当然最好能够保持对称性和正定性，以保证方向是下降的。

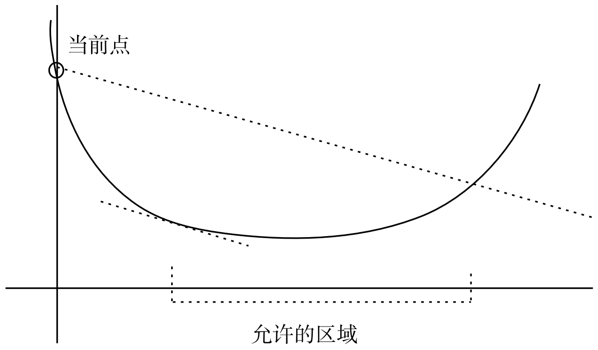


图 21-11 Armijo-Goldstein 条件

一种确保全局收敛的方法要求：f 值能够保证减少一定的步长，步长要足够长，并且搜索方向保持远离与梯度垂直的方向。满足上述要求的一种流行方法是通过 Armijo-Goldstein 条件 [52]，见图 21-11。

(1)

$$f(x_c+\lambda_c p)\leqslant f(x_c)+\alpha\lambda_c\nabla f(x_c)^{\mathrm{T}}p$$

其中 $\alpha\in(0,1)$ 且 $\lambda_c>0$。

(2)

$$\nabla f(x_c+\lambda_c p)^{\mathrm{T}}p\geqslant\beta\nabla f(x_c)^{\mathrm{T}}p$$

其中 $\beta\in(0,1)$。

如果在每次迭代中都满足 Armijo-Goldstein 条件，并且误差有下界，那么有以下**全局收敛性**：

$$\lim_{k\to\infty}\nabla f(x_c)=0$$

前提是每一步都远离垂直于梯度的方向：

$$\lim_{k\to\infty}\nabla f(x_c)s_k/\|s_k\|\neq 0$$

如果 Armijo-Goldstein 条件维持不变，可以使用不失全局收敛的**快速近似一维搜索**[42]。

21.4.2　解决不定黑塞矩阵

如果黑塞矩阵是不定的，可以使用**修正楚列斯基**方法。考虑谱分解：

$$\boldsymbol{H}=\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}}=\sum_{i=1}^{n}\eta_i u_i u_i^{\mathrm{T}}$$

其中 $\boldsymbol{\Lambda}$ 是对角矩阵，$\boldsymbol{\Lambda}_{ii}$ 是本征值 η_i。

很容易可以看出，如果 η_i 是负值（不存在最小值，值可以趋近负无穷大）或者接近零（逆矩阵将有接近无穷大的本征值），将会发生什么。

如果 $\boldsymbol{H}$ 是非正定的或者病态的，非常直接的补救措施之一是加上一个简单的对角矩阵

$$\boldsymbol{H}'=\nabla f(x_c)+\mu_c\boldsymbol{I},\ \ \mu_c>0$$

使得黑塞矩阵 $\nabla^2 f(x_c)+\mu_c\boldsymbol{I}$ 是正定且良置的 ①。

这将引出**修正楚列斯基分解**：找到另一个矩阵 $\bar{\boldsymbol{H}}_c$ 的楚列斯基分解，它们的区别只在于非负对角矩阵 $\boldsymbol{K}$：

$$\bar{\boldsymbol{H}}_c=\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^{\mathrm{T}}=\boldsymbol{H}_c+\boldsymbol{K}$$

① 即非病态的。——译者注

其中，$\boldsymbol{D}$ 中的所有元素都是正的，$\boldsymbol{L}$ 中的所有元素都是一致有界的：

$$d_k > \delta,\ \ |l_{ij}|\sqrt{d_k} \leqslant \beta$$

参考文献 [48] 会告诉你如何选择合适的 β。修正楚列斯基分解使得黑塞矩阵 $\nabla^2 f(x_c) + \mu_c \boldsymbol{I}$ 是正定且良置的 [42,5]。

这相当于在我们的原始模型中增加了一个正定二次型。这使得大步长往往会受到惩罚。

21.4.3 与模型信赖域方法的关系

以前的技术基于寻找一个*搜索方向*，并朝着该方向移动可接受数量的步长（“基于步长的方法”）。

由于上一个修正为此局部模型增加了二次项：

$$m_{\text{modified}}(x_c + s) = m_c(x_c + s) + \mu_c s^{\mathrm{T}} s$$

一个可能的猜想是，最小化这个新模型等价于最小化原有模型，并且约束 s 的步长不能太大。

实现方法可以是首先选择最大步长，然后使用完整（而不是一维）二次模型来确定合适的方向。在**模型信赖域方法**（model-trust region method）中，模型只在某个区域是可信的，这一区域可以通过使用搜索过程中积累的经验来更新。

定理 3 假设寻找步子 s_c，来求解

$$\min m_c(x_c + s) \quad = \quad f(x_c) + \nabla f(x_c)^{\mathrm{T}} s + \frac{1}{2} s^{\mathrm{T}} \boldsymbol{H}_c s \tag{21.12}$$

$$\text{使服从} \|s\| \leqslant \delta_c \tag{21.13}$$

上述问题的解是：

$$s(\mu) = -(\boldsymbol{H}_c + \mu \boldsymbol{I})^{-1} \nabla f(x_c) \tag{21.14}$$

对于使得这一步长是最大允许值（$\|s(\mu)\| = \delta_c$）的唯一的 $\mu \geqslant 0$，牛顿法的步子是问题的解，除非对应于 $\mu = 0$ 的步子在信赖域（$\|s(0)\| \leqslant \delta_c$）里 c 即 $s(0)$ 的情况。

黑塞矩阵的对角线修正是**梯度下降法和牛顿法之间的折中**：当 μ 趋于零时，原来的黑塞矩阵（几乎）是正定的，并且步子与牛顿法的步子趋于重合；当 μ 变大时，对角线的加成 $\mu \boldsymbol{I}$ 趋于主导地位，并且步子趋向正比于负梯度：

$$s(\mu) = -(\boldsymbol{H}_c + \mu \boldsymbol{I})^{-1} \nabla f(x_c) \approx -\frac{1}{\mu} \nabla f(x_c)$$

这些没有必要从一开始就定下来，该算法以自适应的方式选择移动，以适应于误差表面的局部构造。

21.4.4 割线法

如果黑塞矩阵无法计算或者计算成本很高，那么割线法就可以派上用场了。

在一维的情况中，二阶导数可以由两个邻近点的一阶导数值割线的斜率来逼近：

$$\frac{\mathrm{d}^2 f(x)}{\mathrm{d}x^2}(x_2 - x_1) \approx \left(\frac{\mathrm{d}f(x_2)}{\mathrm{d}x} - \frac{\mathrm{d}f(x_1)}{\mathrm{d}x}\right) \tag{21.15}$$

在高维的情况中，如果只有一个方程式，那么是不充分的。令当前点和下一个点分别为 x_c 和 x_+，然后定义 $s_c = x_+ - x_c$ 和 $y_c = \nabla f(x_+) - \nabla f(x_c)$（梯度差）。相应的“割线方程”是：

$$\boldsymbol{H}_+ s_c = y_c \tag{21.16}$$

上述方程并不能决定唯一的 $\boldsymbol{H}_+$，而是从 $(n^2 - n)$ 维仿射子空间 $Q(s_c, y_c)$ 中可以自由选择服从方程 (21.16) 的矩阵。

一个可能的解决问题的方法是使用以前的“历史”。换句话说，方程 (21.16) 不是用来决定，而是用来*更新*先前可用的逼近。

特别是布罗伊登（Broyden）法中，可以使用*最小改变*原则，找到 $Q(s_c, y_c)$（“商”）中最接近先前可用的矩阵的那个矩阵。这是通过将这个矩阵以弗罗贝尼乌斯范数投影到 $Q(s_c, y_c)$ 得到的（矩阵作为一个长向量）。

布罗伊登更新的结果如下：

$$(\boldsymbol{H}_+)_1 = \boldsymbol{H}_c + \frac{(y_c - \boldsymbol{H}_c s_c) s_c^{\mathrm{T}}}{s_c^{\mathrm{T}} s_c} \tag{21.17}$$

然而，布罗伊登更新不保证矩阵是*对称的*（记住，我们想要的是*下降的*方向）。

将布罗伊登矩阵投影到对称矩阵子空间是不够的，得到的矩阵可能不在 $Q(s_c, y_c)$ 中。

幸运的是，如果重复使用上述两种投影，得到的序列 $(\boldsymbol{H}_+)_t$ 将会收敛到一个矩阵，既是对称的，又在 $Q(s_c, y_c)$ 中。这就是鲍威尔（Powell）对称割线更新：

$$\boldsymbol{H}_+ = \boldsymbol{H}_c + \frac{(y_c - \boldsymbol{H}_c s_c) s_c^{\mathrm{T}} + s_c (y_c - \boldsymbol{H}_c s_c)^{\mathrm{T}}}{s_c^{\mathrm{T}} s_c} - \frac{< y_c - \boldsymbol{H}_c s_c, s_c > s_c s_c^{\mathrm{T}}}{(s_c^{\mathrm{T}} s_c)^2} \tag{21.18}$$

现在接近一个令人满意的更新了，但是我们需要的黑塞矩阵的逼近应该是*正定的*。黑塞矩阵 $\boldsymbol{H}_+$ 是对称且正定的，当且仅当 $\boldsymbol{H}_+ = \boldsymbol{J}_+ \boldsymbol{J}_+^{\mathrm{T}}$，其中 $\boldsymbol{J}^+$ 是某个非奇异矩阵。使用布罗伊登法得到一个合适的 $\boldsymbol{J}_+$，从而能够得到一个合适的更新。

这样得到的更新在历史上称为 BFGS 更新 [30]，其中 BFGS 代表布罗伊登（Broyden）、弗莱彻（Fletcher）、戈德法布（Goldfarb）与香农（Shanno），由下式给定：

$$\boldsymbol{H}_+ = \boldsymbol{H}_c + \frac{y_c y_c^{\mathrm{T}}}{y_c^{\mathrm{T}} s_c} - \frac{\boldsymbol{H}_c s_c s_c^{\mathrm{T}} \boldsymbol{H}_c}{s_c^{\mathrm{T}} \boldsymbol{H}_c S_c} \tag{21.19}$$

正定割线更新以 *q-超线性收敛*[30]。

可以用单位矩阵作为初始矩阵 $\boldsymbol{H}_0$，这样第一步是沿着负梯度进行的。

21.4.5 缩小差距：二阶方法与线性复杂度

精确计算黑塞矩阵需要阶为 $O(n^2)$ 的操作数和阶为 $O(n^2)$ 的存储器，用于存储黑塞矩阵的元素，另外求解方程来找到牛顿法的步子（或搜索方向，见图 21-10）需要 $O(n^3)$ 个操作，至少采用传统的线性代数方法时是这样。幸运的是，一些二阶信息可以从前面的梯度开始计算，因此将搜索方向的计算量和内存需求减至 $O(n)$。参考文献 [42] 中的术语“割线法”会让人想起用连接两个函数值的割线来逼近导数。

从历史的角度来说，单步正割法（one-step-secant method，OSS）是所谓单步（无记忆）BFGS 法的变型，见参考文献 [97]。**OSS 方法**已被用于参考文献 [4] 和参考文献 [12] 中的多层感知器。其主要过程示于图 21-12 和图 21-13。

注意，BFGS（见参考文献 [112]）存储整个近似的黑塞矩阵，而单步法**仅需**从梯度计算**向量**。事实上，新的搜索方向的 p_+ 由下式计算得来：

$$p_+ = -g_c + A_c s_c + B_c y_c \tag{21.20}$$

其中的两个标量 A_c 和 B_c 是下列先前定义的向量 s_c、g_c 和 y_c（最后一步，梯度和梯度的差）的标量积的组合：

$$A_c = -\left(1 + \frac{y_c^{\mathrm{T}} y_c}{s_c^{\mathrm{T}} y_c}\right)\frac{s_c^{\mathrm{T}} g_c}{s_c^{\mathrm{T}} y_c} + \frac{y_c^{\mathrm{T}} g_c}{s_c^{\mathrm{T}} y_c} \quad ; \quad B_c = \frac{s_c^{\mathrm{T}} g_c}{s_c^{\mathrm{T}} y_c}$$

搜索方向在学习开始时是负梯度，每隔 N 步后重新变为 $-g_c$（N 是网络中的权重数）。

方向为 p_c 的快速一维最小化，对于获得高效的算法是至关重要的。图 21-13 描述了算法（源于参考文献 [42]）的这一部分。一维搜索基于回溯策略。我们增加上一次成功的学习率 λ（$\lambda \leftarrow \lambda \times 1.1$），并执行第一个试探性的步骤。为了与图 21-12 及图 21-13 的记号相同，使用 E（“能量”）表示需要优化的函数。如果新的 E 值不低于“上限”曲线，那么尝试一个新的试探性步骤，使用连续的二次插值，直到符合要求。注意，每次失败的尝试之后，学习率会下降到 L_{decr}。二次插值不浪费计算资源。事实上，在第一次尝试后，我们正好拥有了拟合一条抛物线所需的信息：初始点的值 E_0 和 E_0'，以及尝试点的值 E_λ。抛物线 $P(x)$ 是：

$$P(x) = E_0 + E_0' x + \left[\frac{E_\lambda - E_0 - \lambda E_0'}{\lambda^2}\right] x^2 \tag{21.21}$$

并且最小值 $\lambda_{\min}$ 是：

$$\lambda_{\min} = \frac{-E_0'}{2\left[\frac{E_\lambda - E_0 - \lambda E_0'}{\lambda^2}\right]} \leqslant \frac{1}{2(1-G_{\mathrm{decr}})}\lambda \tag{21.22}$$

如果图 21-12 中的“梯度乘数”G_{decr} 为 0.5，那么最小化抛物线的 $\lambda_{\min}$ 小于 λ。

ϵ　学习率

$\bar{\epsilon}$　平均学习率

$\boldsymbol{w}_{\text{curr}}$　权重

$\boldsymbol{d}$　搜索方向

```
procedure oss_minimize
    begin_or_restart
    ϵ ← 10^-5
    ϵ̄ ← 10^-5
    w_curr ← 随机初始权重
    迭代 ← 1
    while 未满足收敛条件
        if 迭代次数是 N 的倍数
            begin_or_restart
        迭代 ← 迭代 + 1
        d ← find_search_direction
        if fast_line_search (d) = false                式 (21.20)
            begin_or_restart
procedure begin_or_restart
    找到当前的能量值
    ϵ ← ϵ̄
    d ← −g
    fast_line_search (d)
```

图 21-12　单步割线算法的第一部分[4]

21.5　不涉及导数的技术：反馈仿射振荡器

在现实世界中，存在很多偏导数不能使用的情况，原因是该函数是不可微的，或者是计算导数的成本太高了，因此需要研究仅基于函数值的优化技术，例如基于参考文献 [101] 理论框架基础上的**自适应随机搜索**算法的变种。

通常方案始于在配置空间中选择一个初始点及其周围的初始搜索区域，并按以下步骤进行迭代。

(1) 根据给定的概率度量，在**搜索区域中抽样**，产生一个新的候选点。

(2) **搜索区域**根据新点的函数值**进行调整**。若新的函数值大于当前的（不成功的抽样）值，则缩小搜索区域，反之则扩大搜索区域。

(3) 若抽样是成功的，则新的点成为当前点，并随之**移动搜索区域**，使得当前点成为下一次迭代的中心。

为了得到有效的实现，在当前点周围的简单区域进行搜索就足够了，例如盒形的区域（即区域边缘由一组线性无关矢量给定）并且概率在盒内均匀分布。这种情况下，产生一个随机位移很简单：基础向量乘以范围 $(-1.0, 1.0)$ 内的随机数并加上 $\delta = \sum_j \text{Rand} \times b_j$。

我们可以不严格要求盒形边缘平行于坐标轴，通过沿任意方向的仿射变换，边框可以被压缩或扩展，下一节将进行说明。

21.5.1　RAS：抽样区域的适应性

一个简单但十分有效的，不涉及导数的自适应方法是**反馈仿射振荡器**（RAS）算法 [31]，它基于参考文献 [14]。RAS 通过一个仿射变换来适应搜索区域。两个向量空间的一个仿射变换（来源于拉丁语 affinis,“连接”的意思）由一个紧接平移的线性变换组成：

$$x \mapsto Ax + b$$

在几何学中，欧几里得空间中一个仿射变换保持：

(i) 点之间的共线性（collinearity），也就是说，同一直线上的点在变换后仍在同一直线上；

(ii) 同一直线上距离的比值，也就是说，对于 3 个互不相同的共线点 $p_1$1、$p_2$2、$p_3$3，比值 $|p_2 - p_1|/|p_3 - p_2|$ 将会保持不变。通常，一个仿射变换是由线性变换（旋转、缩放或剪切）和一个平移变换（或者“转移”）组成。

在 RAS 中，当找到一个成功的样本时，区域沿任意成功方向拉长；如果没有找到，区域沿失败方向压缩。这些修改考虑到均匀分布生成的尝试点给出的*局部信息*。这样做的目的是在包含初始点的吸引域（attraction basin）中搜索局部极小值，通过调整步长大小和方向，以求在**每次进行函数求值之后都启发式地保持最大可能的移动**。这种设计的补充是某些策略选择的分析，如双射策略（double-shot strategy）和初始化 [31]。现在谈一谈这个方法的名称（反馈仿射振荡器）。该解决方案尽量减少跳向最小区域的跳跃数目，这是由不断变化移动方向和大小来实现的。搜索区域和步子的调整因此是通过由搜索本身的演变引导的反馈回路实现的，也就是实现一个“反馈”自我调节的机制。生成抽样的过程根据 f 表面的*局部特性*进行调整，引导它的是反馈搜索优化原则的灵魂，这将在第 22 章讨论。步长和方向的不断变化创造了一个“摇摇欲坠”的轨迹，有着突然性的跳跃和转折。

RAS 算法的伪代码见图 21-14。每次迭代时，会产生一个位移 $\boldsymbol{\Delta}$，使得点 $\boldsymbol{x} + \boldsymbol{\Delta}$ 均匀分布在局部搜索区域 $\mathcal{R}$ 中（第 4 行）。为了达到这个目的，基向量都被乘以了范围在 $[-1, 1]$ 的不同的随机数，并且加上：

$$\boldsymbol{\Delta}=\sum_j \mathrm{Rand}(-1,1)\boldsymbol{b}_j$$

Rrand$(-1,1)$ 表示调用随机数发生器。如果 $\boldsymbol{x}+\boldsymbol{\Delta}$ 或者 $\boldsymbol{x}-\boldsymbol{\Delta}$ 中的某一个改进了函数值，那么它被选择为下一个点。把这个改进点记为 $\boldsymbol{x}'$。为了沿着有希望的方向扩展这个盒形区域，这些盒型向量 $\boldsymbol{b}_i$ 需要做如下修改。

改进的方向是 $\boldsymbol{\Delta}$。将相应的归一化为单位长度的向量记为 $\boldsymbol{\Delta}'$：

$$\boldsymbol{\Delta}'=\frac{\boldsymbol{\Delta}}{\|\boldsymbol{\Delta}\|}$$

那么，向量 $\boldsymbol{b}_i$ 沿着 $\boldsymbol{\Delta}$ 方向的投影是：

$$\boldsymbol{b}_i|_{\boldsymbol{\Delta}}=\boldsymbol{\Delta}'(\boldsymbol{\Delta}'\cdot b_i)=\boldsymbol{\Delta}'\boldsymbol{\Delta}'^{\mathrm{T}}\boldsymbol{b}_i$$

为了获得所需的效果，这一部分以某个系数 $\rho>1$ 扩大，所以新向量 $\boldsymbol{b}_i'$ 的表达式是：

$$\begin{aligned}\boldsymbol{b}_i' &= \boldsymbol{b}_i+(\rho-1)b_i|_{\boldsymbol{\Delta}} \\ &= \boldsymbol{b}_i+(\rho-1)\boldsymbol{\Delta}'\boldsymbol{\Delta}'^{\mathrm{T}}\boldsymbol{b}_i \\ &= \boldsymbol{b}_i+(\rho-1)\frac{\boldsymbol{\Delta}\boldsymbol{\Delta}^{\mathrm{T}}}{\|\boldsymbol{\Delta}\|^2}\boldsymbol{b}_i \\ &= P\boldsymbol{b}_i\end{aligned} \tag{21.23}$$

其中

$$P=\boldsymbol{I}+(\rho-1)\frac{\boldsymbol{\Delta}\boldsymbol{\Delta}^{\mathrm{T}}}{\|\boldsymbol{\Delta}\|^2} \tag{21.24}$$

测试函数值在两个点 $\boldsymbol{x}+\boldsymbol{\Delta}$ 和 $\boldsymbol{x}-\boldsymbol{\Delta}$ 上的改进被称为*双射策略*：如果第一样本 $\boldsymbol{x}+\boldsymbol{\Delta}$ 不成功，就考虑对称点 $\boldsymbol{x}-\boldsymbol{\Delta}$。这样的选择大大减少了产生两个连续不成功样本的概率。为了解释起来更具体，考虑围绕当前点拟合一个*平面*：如果某一步增加了 f，那么相反的一步会减少 f。转到数学层面，如果考虑可微函数和小位移，沿位移方向的导数与位移和梯度 $\boldsymbol{\Delta}\cdot\nabla f$ 之间的标量积是成比例的。如果第一项为正，那么符号的变化就会产生无用的负值，因此对于足够小的步长，f 也会减小。对于一般的不一定可微的函数，实践证明也是有效的，这是由于很多对应于现实世界问题的函数所包含的相关性和结构。

如果双射策略失败，那么运用仿射变换 [式 (21.23)] 时，膨胀系数 ρ 就由它的逆 ρ^{-1} 替代（如图 21-14 第 12 行所示），使得搜索区域缩小。

图 21-15 展示了反馈仿射振荡器算法的几何方面，其中需要被最小化的函数由固定 f 值的等高线表示，并且画出了两个轨迹（ABC 和 $A'B'C'$）。沿着搜索轨迹的一些点展示了搜索区域。RAS 的设计准则由*局部极小值的激进搜索*给定：当搜索步子（点 A 和 A'）成功时，搜索速度加快；如果在双射之后没有发现更好的点，搜索速度减慢。当一个点接近局部极小值

时，不断减少搜索框将产生一个收敛非常快的搜索（C 点）。注意，缩小搜索区域的另一个原因可能是一条狭窄的下降通路（一个“峡谷”，例如 B' 点），这时，所有可能的方向上只有一小部分能改进函数值。然而，一旦发现了一个改进，搜索区域就应该沿着这个有希望的方向扩展，从而能够在该方向上更快地移动。

21.5.2 为健壮性和多样化所做的重复

在一般情况下，有效估计全局极小值的步数显然是不可能的。即使找到了局部极小值，一般不可能确定它是否为全局最优，特别是如果只是从 $f(x)$ 在所选的点上的值得到关于该函数的知识，而这是处理现实世界问题的一种常见情况。

因为 RAS 不包括避免局部极小的机制，所以当轨迹足够接近一个局部极小时，就应该停止搜索。例如，如果搜索区域小于某个阈值，RAS 的单次运行就应该被终止。事实上，如果在附近重复改进函数值而未能成功，盒区域往往会减小其体积至接近局部极小值。

RAS 搜索局部极小，只要找到一个就停止搜索。让搜索继续的一个简单方法是**从不同的初始随机点重新开始运行**。这种方法相当于有一个 RAS 搜索器的“总体”样本，这个总体中的每个成员是*独立的*，完全不知道其他成员正在做什么（图 21-16）。第 23 章将会讨论更复杂的协调搜索器团队的方式：CoRSO 框架。

通过学习 RAS 的经验得出的一个建议是：**懒惰！先尽量尝试简单的方法，仅当期待一个可观的改进时，才增加复杂度**（KISS 原则）。注意，RAS 需要矩阵和向量的相乘来更新搜索区域，因此当维数变得非常大时，这一方法会变得缓慢。图 21-17 中概述了更简单的**惯性振荡器**（inertial shaker）技术，在维数很大的时候可能是一个更有效的选择：搜索框总是由平行于坐标轴的向量（因此搜索框由单个向量 $\boldsymbol{b}$ 确定，并且不需要矩阵乘法）和一个*趋势方向*确定，该趋势方向由许多之前的位移的平均值确定 [14]：第 7 行中的 find_trend 函数只是返回先前位移的加权平均值 m_{disp}：

$$\boldsymbol{\delta}_t = \text{amplification} \cdot \frac{\sum_{u=1}^{\mathrm{T}} \boldsymbol{\delta}_{t-u} \mathrm{e}^{-\frac{u}{(\text{history_depth})^2}}}{\sum_{u=1}^{\mathrm{T}} \mathrm{e}^{-\frac{u}{(\text{history_depth})^2}}}$$

其中 amplification 和 history_depth 由算法所定义，而 m_{disp} 的选择是用来去除可忽略不计的指数权重，使得历史记录保持合理的长度。

图 21-18 显示了在搜索位置 $\boldsymbol{x}$ 处如何将双射策略应用到所有组件。只要改进了结果，就对每个部件都施加一个位移。如果没有改进的可能，那么函数返回 `false`，并将搜索框相应缩小。

$\boldsymbol{d}$	搜索方向
$\boldsymbol{g}$	函数梯度
$\boldsymbol{w}$	权重
d_l	$\boldsymbol{d}$ 在梯度方向上的投影
E	当前能量
E_{saved}	最优能量
ok	发现可提升步骤
$trials$	迭代次数
$MAXTRIALS$	允许的最大迭代次数
L_{decr}	每次迭代下降的步数

```
procedure fast_line_search (d)
  d_l ← g · d
  if d_l > 0
     d ← - g ; d_l ← g · d
  E_saved ← E
  ε ← L_incr ε ; ok ← false ; trials = 0
  repeat
     trials ← trials + 1
     w ← w_curr + ε d
     E ← E(w)
     if E < E_saved + G_decr d_l ε
        ok ← true
     else
        ε_quad ← parabola_minimizer (E_saved , d_l ,f)          式 (21.22)
        w ← w_curr + ε_quad d
        E ← E (w)
        if E < E_saved + G_decr d_l ε_quad
           ok ← true ; ε ← ε_quad
        else
           ε ← L_decr ε
  until ok = true or trials > MAXTRIALS
  if ok = true
     p ← ε d
     w_curr ← w
     g ← ∇_w E (w)
     ε̄ ← 0.9 ε̄ + 0.1 ε
  return ok
```

图 21-13 单步割线算法的第二部分 [4]：在所选方向上快速进行一维搜索

f	（输入）	要最小化的函数
$\boldsymbol{x}$	（输入）	初始点
$\boldsymbol{b}_1, \cdots, \boldsymbol{b}_d$	（输入）	表示 $\boldsymbol{x}$ 周围搜索域 $\mathcal{R}$ 的向量
ρ	（输入）	盒形拓展因子
t	（内部参数）	迭代次数
$\boldsymbol{P}$	（内部参数）	变换矩阵
$\boldsymbol{x}$，Δ	（内部参数）	当前值和位移

```
function ReactiveAffineShaker(f, x, (b_j), ρ)
    t ← 0
    repeat
        Δ ← Σ_j Rand(−1,1) b_j
        if f(x + Δ) < f(x)
            x ← x + Δ
            P ← I + (ρ − 1) ΔΔ^T / ‖Δ‖²
        else if f(x − Δ) < f(x)
            x ← x − Δ
            P ← I + (ρ − 1) ΔΔ^T / ‖Δ‖²
        else
            P ← I + (ρ^{−1} − 1) ΔΔ^T / ‖Δ‖²
        ∀j b_j ← P b_j
        t ← t+1
    until 满足收敛条件
    return x
```

图 21-14 RAS 算法的伪代码

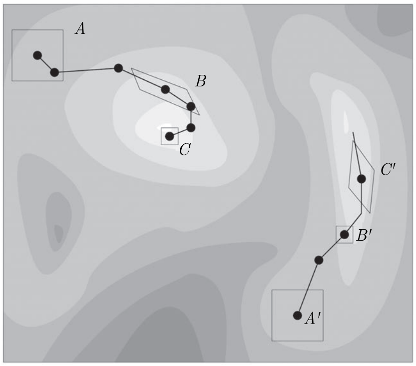


图 21-15 RAS 的几何示意图：两条搜索轨迹得到两个不同的局部极小值，改编自参考文献 [31]

f	（输入）	要最小化的函数
ρ	（输入）	盒形拓展因子
$L_1, \cdots, L_d, U_1, \cdots, U_d$	（输入）	搜索范围
$L'_1, \cdots, L'_d, U'_1, \cdots, U'_d$	（输入）	初始范围
$\boldsymbol{b}_1, \cdots, \boldsymbol{b}_d$	（内部参数）	表示 $\boldsymbol{x}$ 周围搜索域 $\mathcal{R}$ 的向量
$\boldsymbol{x}, \boldsymbol{x}'$	（内部参数）	当前值，最终运动值

function RepeatedReaciveAffineShaker (f, ρ, (L'_j), (U'_j), (L_j), (U_j))
  $\forall j\ \boldsymbol{b}_j \leftarrow \frac{U_j - L_j}{4} \cdot \boldsymbol{e}_j$
  pardo
    $\boldsymbol{x} \leftarrow$ 随机值 $\in [L'_1, U'_1] \times \cdots \times [L'_d, U'_d]$
    $\boldsymbol{x}$' ← ReactiveAffineShaker(f, $\boldsymbol{x}$, ($\boldsymbol{b}_j$), ρ)
  return 找到的最优值

图 21-16 RAS 算法，来自参考文献 [31]

f	（输入）	要最小化的函数
$\boldsymbol{x}$	（输入）	初始值和当前值
$\boldsymbol{b}$	（输入）	表示 $\boldsymbol{x}$ 周围搜索域 $\mathcal{R}$ 的盒形区域
$\boldsymbol{\delta}$	（参数）	当前偏移
amplification	（参数）	未来位移的放大因子
history_depth	（参数）	历史位移均值的权重衰减因子

function InertialShaker (f, $\boldsymbol{x}$, $\boldsymbol{b}$)
  $t \leftarrow 0$
  重复
    success ← double_shot_on_all_components ($\boldsymbol{\delta}$)
    if *success* = `true`
      $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{\delta}$
      find_trend ($\boldsymbol{\delta}$)
      if $f(\boldsymbol{x} + \boldsymbol{\delta}) < f(\boldsymbol{x})$
        $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{\delta}$
        increase amplification 和 history_depth
      else
        decrease amplification 和 history_depth
  直到满足收敛条件
  return $\boldsymbol{x}$

图 21-17 惯性振荡器算法，来自参考文献 [14]

f	要最小化的函数
$\boldsymbol{x}$	当前值
$\boldsymbol{b}$	表示当前搜索盒形区域的向量
$\boldsymbol{\delta}$	位移

```
function double_shot_on_all_components (f, x, b, δ)
    success ← false
    x̂ ← x
    for i ∈ {1, ···, n}
        E ← f(x̂)
        r ← [−b_i, b_i] 中的随机值
        x̂_i ← x̂_i + r
        if f(x̂) > E
            x̂_i ← x̂_i − 2r
            if f(x̂) > E
                b_i ← ρ_comp b_i
                x̂_i ← x̂_i + r
            else
                b_i ← ρ_exp b_i
                success ← true
        else
            b_i ← ρ_exp b_i
            success ← true
    if success = true
        δ ← x̂ − x
    return success
```

图 21-18 参考文献 [14] 中的双射策略：对所有的组件在搜索框内施加一个随机位移，保证每一步都有改进，反之返回 false



梗　概

对实数作为输入参数的函数（模型）进行优化是一个古老的领域，大约始于第二次世界大战期间，现在发展到了很高的水平。这一研究的目的在于设计**自动化技术来找到导致最大（或最小）输出值的输入**。

尽管数学是复杂的，大多数优化技术的基础还是非常容易理解的，哪怕你不记得在微积分（研究变化的数学）课堂上学了什么。天上掉下来的一滴水融入大海，这一过程无须任何数学技巧。

优化技术基本步骤如下。从输入参数的初始值开始，对各种输入加入小的局部变化并测试它们的影响（看看是否导致更大或更小的输出值）。根据测试结果，决定是否接受该局部变化。重复这一过程，直到有所改进，从而获得越来越好的输出值。

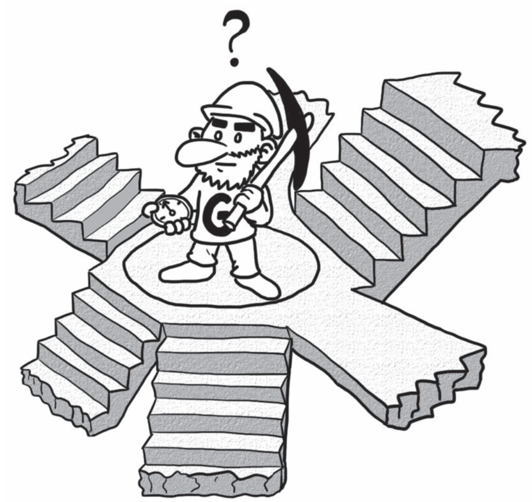
如果能计算导数，就有了简单的方法来预测小的局部变化的影响。事实上，可以把导数当作变化的局部预测。如果该步骤足够小，以至于“变化等于导数乘以步骤”，那么它的近似也会比较理想。如果导数不可用，那么可以直接测试微小变化（例如在 RAS 中），并不断**适应局部模型**来减少无用的的函数求值。局部适应性需要**从前面的搜索步骤中进行学习**。

如果能理解原理，即使没有数学定理，也足够让你自如地使用优化软件，并能够避开大多数的陷阱。毕竟，你滑雪的时候，不需要微积分和数学分析来保证你不会摔倒，或者保证你到达滑雪缆车。

第 22 章　局部搜索和反馈搜索优化

每个人都肩负着各自的选择，这也不是难以承受之重。

——-Romano Battiti



现在来考虑寻找离散（组合）的极小值的最优化问题。例如，给定城市和相互之间距离的列表，想要找到恰好经过每个城市一次的最短路线。这就是所谓的**旅行商问题**（TSP），该问题和旅行的相关性是显而易见的，比如可以减少路费和二氧化碳排放量。TSP 是运筹学和理论计算机科学研究中的一个非常棘手的问题。它最早作为一个数学问题在 1930 年提出，如今是优化领域研究最深入的问题之一。找到大型实例的最优解在计算上是困难的（实际上在大多数情况下，这是不可能的），但人们依旧提出了很多启发式的算法，即使实例中的城市有成千上万座，在实践中都能够有效地解决。

整体上抽象来说，给定一个函数 f，定义在一组离散的可能输入值 $\mathcal{X}$ 上，路线长度定义为 TSP 中城市排列的函数，目标是寻找使得函数 f 取得最小可能值的输入。本章将介绍局部搜索的基本要素，以及更先进的**反馈搜索优化**（RSO），它带有内部在线的自我调整机制。为了避免混淆，注意这里的“局部搜索”与网页信息搜索技术（如谷歌或相似服务）没有任何关系。这里的搜索是为了提升优化问题的结果，如果可能就搜索最优值，否则就搜索近似值。

RSO 原则在连续优化和多目标优化领域的应用将分别在第 23 章和第 24 章讨论。

22.1　基于扰动的局部搜索

一个基本的解决问题的策略，是先从一个试探性的解决方案开始，尝试多次小幅度的修改，从而改进这一解决方案。每次重复时，对当前的构型稍做修改（扰动），并对需要优化的函数进行测试。如果新的解决方案比原来更好，就保留这一变化，否则就再尝试另一个变化。需要优化的函数 $f(X)$ 在某些领域里拥有更富诗意的名字：适应度函数、优度函数、目标函数。

图 22-1 展示了自行车设计史上的一个例子。这里不是为了再现历史，毕竟本书谈论的是 LION 技术，而不是自行车技术。第一种模型是初始模型，只有一个轮胎，它可以工作，但它还不是最佳的。第二种模型尝试随机添加一些组件到原设计中，这下情况会更糟。此时可以恢复到初始模型，并尝试其他变化。值得注意的是，如果坚持在第二种模型上继续添加一些东西，最终可能得到第三个模型，从可用性和安全性角度来看，显然会更胜一筹。这个故事给我们上了一课：**基于微小扰动的局部搜索是一种美味的食材，但某些情况下需要加点香料**让味道更好。



图 22-1　局部搜索现实生活中的一个例子：如何构造一个更好的自行车，从初始模型（左）到一个更糟糕的版本（中），最后是一个更好的版本（右）

另外，值得注意的是，每个人的生活都是优化算法的一个例子：大部分变化是局部的，戏剧性的变化确实会发生，但不会非常频繁。想象一下，你找到了一个伴侣，一个可能伴随你一生的伴侣。这段关系一开始的局部变化可能会很频繁。例如，你可能想要说服你的伴侣打扮更得体，不吃大蒜，或改变对各种问题的看法。为了增进感情，能以更轻松的方式在一起，你可能要忍受一些不好的变化，比如容忍你的伴侣看周末的足球比赛。或者你最终认为这些小的变化于事无补，出路是激烈的多样化或重新开始一段关系（寻找更好的伴侣）。

基于扰动一个候选解决方案的局部搜索，就是可以应用简单学习策略的第一范式情况。接下来定义符号。$\mathcal{X}$ 是搜索空间，$X^{(t)}$ 是在迭代（“时间”）t 的当前解。$N(X^{(t)})$ 是点 $X^{(t)}$ 的

邻域，通过将一系列基本动作 $\mu_0, \cdots, \mu_M$ 应用到当前解而得到：

$$N(X^{(t)}) = \{X \in \mathcal{X} \text{ 使得 } X = \mu_i(X^{(t)}), i = 0, \cdots, M\}$$

如果搜索空间由给定长度 L 的二进制串来确定 $\mathcal{X} = \{0,1\}^L$，那么可以采取的行动是改变（求补或翻转）各个二进制位，因此 M 等于串长度 L。

*局部搜索*从可容许解 $X^{(0)}$ 开始，并确定一个**轨迹** $X^{(0)}, \cdots, X^{(t+1)}$。当前点的后继是邻域中的一个点，需要最小化的函数 f 在这个点上有一个较小的值：

$$Y \leftarrow \text{Improving-Neighbor}(\ N(X^{(t)})\ ) \tag{22.1}$$

$$X^{(t+1)} = \begin{cases} Y & \text{如果 } f(Y) < f(X^{(t)}) \\ X^{(t)} & \text{其他情况（停止搜索）} \end{cases} \tag{22.2}$$

Improving-Neighbor 返回邻域中改进的元素。在简单情况下，它是具有最小 f 值的元素，但也存在其他的可能性，例如遇到的*第一个邻域改进*。

如果没有邻域有更好的 f 值，也就是说，如果解是一个**局部极小**（local minimizer），那么搜索停止。请注意，最大化一个函数 f 等价于最小化 $-f$。像所有对称的情况那样，这其实会造成术语上的一些混乱。例如，*最速下降*假设了最小化观点，而*爬山法*假设了相反观点。这本书大部分将讨论最小化，*局部极小值*（local minimum）是不能通过移动到其邻域进行改进的点。局部最优（local optimum）是既可以用于最大化和又可以用于最小化的术语。

局部搜索出奇地有效，因为大多数组合优化问题都有非常**丰富的**将解 X 和 f 值联系起来的**内部构型**。当输入域是实数 $\mathbb{R}^n$ 时，可以进行类比，连续可微函数 $f(x)$ 通过梯度下降（又名最速下降）优化。顺便提醒一句，当考虑全局收敛时，最速下降不一定是最好的方法：正如第 21 章所解释的，无论是离散还是连续，贪心并不总会赢。

邻域（neighborhood）适合局部搜索，如果它反映了问题的构型。例如，如果解是由一个排列给定的（在旅行商问题中被访问的城市的排列），描述当前解的二进制串中的单个二进制位的变化是一个不恰当的邻域选择，这将立刻导致*非法二进制数*（illegal configuration），这一编码不对应于任何排列。为了得到一个更好的邻域，可以交换解的两个元素，并保持其他所有位固定。一般情况下，如果邻域的 f 值与当前点的 f 值相关，一个全面的邻域检查就能掌握局面。如果起点是个很好的解决方案，平均而言，相比于完全无关的随机点，*邻域中*更容易找到同等质量的解。顺便说一下，抽样一般的随机点比抽样邻域更昂贵，前提是邻域的 f 值可以被更新（**"增量求值"**）并且不必从头重新计算。

许多值得研究的优化问题需要更接近全局最优的逼近，因此需要更复杂的方法，以便继续调查搜索空间的新的部分，即*搜索多样化和探索*。这里另一个构型性因素可以起作用，它涉及局部极小值和相应 f 值的整体分布。在许多相关的问题中，局部极小值往往聚集在一起，而且好的局部极小值往往更接近其他好的极小值。**看上去有希望的局部极小值喜欢待在一起**。下面定义与局部最优相关的**吸引域**，它是通过局部搜索轨迹被映射到某给定的局部最优值的

点集 X。用水来类比，其中局部搜索轨迹就像是水受到重力吸引而留下的轨迹，而吸引域就好比是流域，即由分水线圈定的区域，降水最终流入某一湖泊。

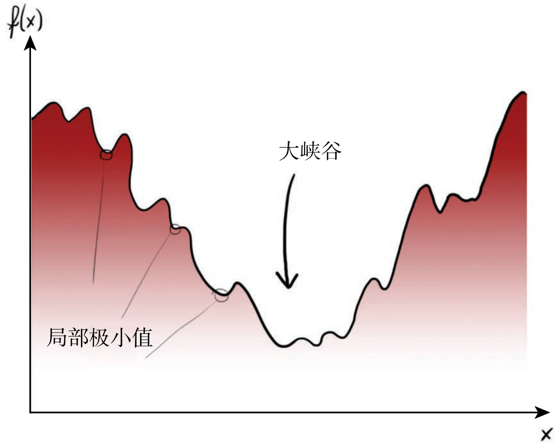


图 22-2　最优化问题中的构型："大峡谷"特性

话说回来，如果局部搜索在一个局部极小值处停止，那么将系统**带到**一个邻近的吸引域可能远比从随机解重新开始更加有效。如果 f 的求值使用增量法，那么经过一系列步骤移动到附近的吸引域，总比重新开始完整的求值，沿着一条可能很长的下降轨迹来到另一局部最优要快得多。

这种构型特性也称为**大峡谷特性**（见图 22-2）。为了更直观，可以想象在连续的环境中有一光滑的 f 表面，带有一些往往是嵌套、"分形"构型的吸引域。根据芒德布罗的定义，分形一般是指"粗糙或零碎的、可被细分成更小的部分的几何形状，其中每一部分都是（至少近似）整体的小尺寸的复制"，该属性也称为自相似性 ①。

第二个连续的类比是用傅里叶变换来分析含有不同波长分量的（周期性）函数。如果你不是傅里叶变换的专家，想想通过散焦镜头来看某幅画。起初会看到一些较粗略的细节，例如一个远处的人影，然后通过聚焦，就能看到越来越多的细节：脸、胳膊和腿，然后是手指、头发等。相同的类比也存在于由不同质量的扬声器播放的音乐，扬声器质量越好，能听到的频率就越高。在每个不同的音阶，声音不是由随机噪声和一个模式构成的；一个不平凡的构型总是存在的。这种**多标度**（multi-scale）构型，即小峡谷被嵌套在较大的峡谷里，是可变邻域搜索（Vanable Neighborhood Search，VNS）和迭代局部搜索（Iterated Local Search，ILS）这些方法的基本动机。例如，参考文献 [9] 中的介绍更详尽，也包含对相关技术的讨论，或者参考文献 [8] 中较简略的介绍。

①分形（fractal）一词来源于拉丁语 fractus，意思是"破碎的"或"断裂的"。

22.2 反馈搜索优化：搜索时学习

通过总结参考文献 [7,9,8] 中的拓展主题，我们可以找到从简单的局部搜索跨越到反馈搜索优化的主要动机。

许多解决问题的方法是由一定数目的选择和自由参数来确定的，而适当设置和调整这些选择和参数是复杂的。某些情况下，这些参数通过一个反馈回路进行调整，这个回路**将用户作为学习的关键组件**：开发和测试不同的选项，直至获得可以接受的结果。结果的质量不会自动传输给不同的实例，当该算法必须为了某个特定的应用进行调整时，反馈回路可以要求一个慢速的“试错”过程。机器学习领域有丰富多样的“设计原则”，可用于启发式参数调整领域，开发机器学习方法。这种方式消除了人工干预，但并不意味着研究人员会有较高的失业率。相反，人们现在的任务更重：**算法开发者必须将其智能专长传输给算法**本身，这一任务需要在算法的调整阶段进行详尽描述。算法的复杂性会增加，但如果以下两个目标都能达到，那么这一代价是值得的。

- 完整和明确的文档保证**结果的可重复性**。算法变得自容（self-contained），其质量可以由设计者或特定用户来独立判断。这一要求对科学意义重大，因为在科学中，客观的评价是至关重要的。软件文档的广泛使用进一步简化了测试，启发式算法的再利用也变得更加简单。
- **自动化**。原本耗时的手工调节阶段如今被自动过程所取代，如图 22-4 所示。注意，通常只有最终用户才会从自动调节过程中受益。相反，算法设计者将面临更长、更困难的开发阶段。世上没有免费的午餐：复杂性并没有消失，只是从决策者转移到了方法（和软件）的设计者。

反馈搜索优化（Reactive Search Optimization，RSO）主张将在线机器学习技术融入*启发式搜索*，用以解决复杂的优化问题。“反馈”这个词所暗示的是，为了在线自我调节和动态适应，对于通过内部反馈环路进行的搜索，存在一个即时的反应。在 RSO 中，过去的搜索历史和在解空间中移动所积累的知识，以自动的方式被用于自适应：在搜索过程中，该算法保持解决不同情况所需的内部灵活性，但这种调节是自动的，当算法在单一的实例上运行并使用其过去的经验时执行。因此机器学习是煲 RSO 这碗汤的必要成分，如图 22-3 所示。

反馈搜索优化的一个使用场景是某个系统（电子系统、工厂、卡车车队、业务流程）要求你设置一些工作参数（旋钮、开关、行程计划、程序）以改善其功能。依参数设置而定，系统可能给出更好或更坏的结果（由生产速度、净利润、油耗、客户满意度等来度量）。

为了优化结果，我们执行一个简单的循环：设置参数，观察结果，然后以战略性和智能的方式改变参数，直到找到一个合适的解决方案。该方法的智能性来自 RSO 组件，基于正在进行的搜索过程收集到的信息，该组件决定下一步是什么。为了有效运作，反馈搜索优化使用记忆和智能，以针对性和集中的方式来改进解决方案。反馈搜索优化采用机器学习和统计学

的思路和方法，特别是强化学习，主动或查询学习，以及神经网络。

反馈搜索优化中的隐喻大部分来自人类个体的经验。它的座右铭是“**边干边学**”。如前所述，实际问题有丰富的结构。当我们在搜索空间中测试许多不同的解决方案时，模式和规律就会出现。人类大脑能很快从中学习，并基于先前的观察做出关于未来的决定。这一类比是将在线机器学习技术插入到 RSO 的优化引擎主要的灵感来源。模因算法（memetic algorithm）对于学习有着相似的关注点，不过该算法关注文明的进化，描述社会如何随着时间的推移发展，而不是个体的能力。

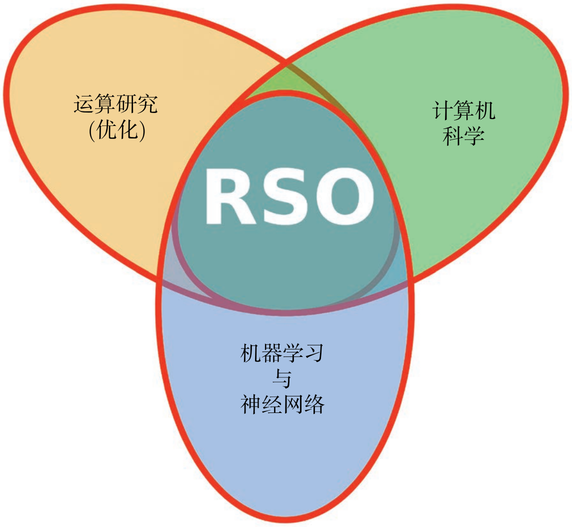


图 22-3　RSO 是优化、计算机科学（算法和数据结构）和机器学习的交集

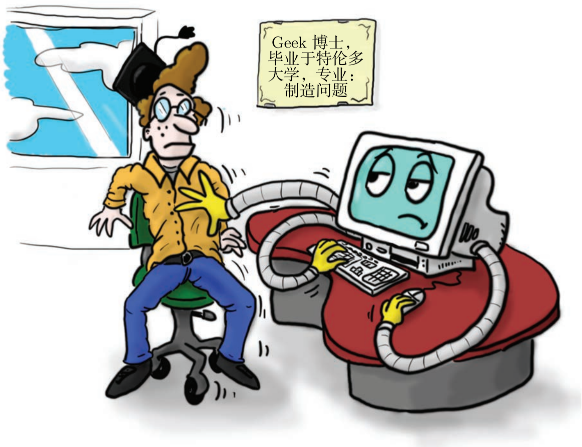


图 22-4　具有自校正（self-tuning）功能的算法（例如 RSO）让最终用户的生活更简单。解决复杂问题不再需要技术专长，而是面向更广大的最终用户群体（改编自参考文献 [9]）

今天，以自然界和生物为灵感的优化算法隐喻比比皆是。某种程度上令人惊讶的是，这些隐喻大部分来自遗传和进化，或者从简单有机体互动涌现出的集体行为，而这些有机体基本都没什么学习能力。这不由得令人猜想是否有关让–雅克 卢梭所持的偏见，他相信自然状态下的人性是好的，但是社会使人变坏；或者这和好莱坞商业电影所秉承的“对抗自然法则的坏人”的原则一脉相承。但隐喻会带我们偏离主方向：我们是实用方法的坚定支持者，**一个算法是有效的，是因为它出色地解决了问题，并且不需要耗费太多精力来调整**，而不是因为它对应于人们喜好的那种精巧、怪诞或引人遐想的类比。此外，至少对于研究人员来说，大多数情况下如果有办法来分析一个算法的行为，并解释它是为何有效以及何时有效，那么它在科学上就是有趣的。弗雷德 格洛弗开创性的论文谈论了禁忌搜索，分散搜索和路径重链，以及相关的元启发式，见参考文献 [49，51] 和有趣的文献 [50]。还有一些令人振奋的文章，你可能想要读读，看一看相关主题，比如模因算法 [82]、进化神经网络 [118]、元启发式 [25]，还有性能预测和自动调节 [66]。

22.3 基于禁忌的反馈搜索优化

参考文献 [9] 阐明，RSO 原理可用于许多算法参数的在线自校正。这一原理最初应用于基于禁止（prohibition）的局部搜索领域，而现在它已成为一个流行术语——禁忌搜索（tabu search）。

使用禁忌以鼓励创新和多样化的想法，即鼓励决策者、工程师或设计师从根本上考虑新替代品的理念，深深根植于搜索的实践中。康拉德 洛伦茨，这位来自澳大利亚的诺贝尔奖得主和现代行为学的创始人说：“*每天早餐前都抛弃一个心爱的假设。这对从事研究的科学家来说是很好的晨练。这使他年轻。*”这句话漂亮地阐述了一个事实：为了拥有真正的创造，必须禁止旧的解决方案。在前面的例子中，人们必须停止考虑独轮车，而这最终促使自行车得以实现！

如上所述，局部搜索在允许的搜索空间中生成*轨迹* $X^{(t)}$。点 X 的后继是从 X 的邻域 $N(X)$ 中选出来的，$N(X)$ 是 $\mathcal{X}$ 的一个子集。如果对于所有的 $Y \in N(X)$，$f(X) \leqslant f(Y)$，那么 X *点相对于* N *是局部最优，或者局部极小*。在下面的讨论中，考虑 $\mathcal{X}$ 包含有限长度 L 的二进制串的情况，即 $\mathcal{X} = \{0,1\}^L$，并且通过把字符串 $X = [x_1, \cdots, x_i, \cdots, x_L]$ 第 i 位进行改变的基本操作 μ_i（$i = 1, \cdots, L$）得到邻域：

$$\mu_i([x_1, \cdots, x_i, \cdots, x_L]) = [x_1, \cdots, \bar{x}_i, \cdots, x_L] \tag{22.3}$$

其中 $\bar{x}_i$ 是第 i 位的否定：$\bar{x}_i \equiv (1 - x_i)$。

RSO 方法使用一个迭代的*修正局部搜索*算法，让搜索偏向 f 值更小的点，它包含了反馈禁忌策略（reactive prohibition strategy），不鼓励重复访问构型。在迭代的每一步，选择的搜索指向在邻域内使得损失函数 f 值最小的点。即使 f 关于当前点的值是递增的，也需要移动，

继而远离 f 的局部极小值。一旦某个移动被应用，那么它的逆向移动暂时会被禁用（“禁忌”就源于这样的禁用操作）。

进一步来讲，在一个给定的搜索循环 t 中，移动的集合 $\mathcal{M}$ 被划分为禁忌集合 $\mathcal{T}^{(t)}$ 和准许集合 $\mathcal{A}^{(t)}$，带括号的上标表示其所处的搜索循环。最初，搜索开始于一个起始的构型 $X^{(0)}$，该构型是随机生成的，且所有的移动路径都是允许的：$\mathcal{A}^{(0)}=\mathcal{M}$, $\mathcal{T}^{(0)}=\emptyset$。然后通过应用集合 $\mathcal{A}^{(t)}$ 中最佳的准许移动 $\mu^{(t)}$，生成搜索的路径 $X^{(t)}$：

$$X^{(t+1)}=\mu^{(t)}(X^{(t)}) \quad \text{其中} \quad \mu^{(t)}=\arg\min_{\nu\in\mathcal{A}^{(t)}} f(\nu(X^{(t)}))$$

孤立地看，这个“修正贪心搜索”原则能生成循环（cycle）。例如，如果现在这一点上的 $X^{(t)}$ 是一个严格的局部极小值，那么下一个点的损失函数值一定会变大：$f(X^{(t+1)})=f(\mu^{(t)}(X^{(t)}))>f(X^{(t)})$，而且下一步有可能沿着**反方向**移动（$\mu^{(t+1)}={\mu^{(t)}}^{-1}$），因此沿着上述两步就回到了最开始的点：

$$X^{(t+2)}=\mu^{(t+1)}(X^{(t+1)})={\mu^{(t)}}^{-1}\circ\mu^{(t)}(X^{(t)})=X^{(t)}$$

此时，如果准许的移动集合是相同的，那么这个系统将会永远困在一个长度为 2 的循环中。这个例子中，如果反方向移动 ${\mu^{(t)}}^{-1}$ 在 $t+1$ 次迭代被禁止，那就可以避免这个循环。通常来说，所有**最近一段时间内搜索移动的反方向都应该在一定时间段 T 内被禁止**。如果用二进制串表示一个移动及其反方向移动，就是一个移动被禁止，当且仅当它在最近的 $\tau\geqslant(t-T^{(t)})$ 时间内被执行过。这个时间段是有限的，因为禁止的移动对于后续搜索最优值的过程是必要的。在 RSO 中，禁止时间段 $T^{(t)}$ 与迭代时间 t 有关。

搜索轨迹上禁止移动的多样化影响，已经在**禁忌和多样化的基本关系**中得到了解释，并在参考文献 [18] 中得到了证明。令 $H(X,Y)$ 为两个二进制串 X 和 Y 之间的汉明距离，定义为 X 和 Y 之间不同的二进制位数。现在，假设只能执行准许的移动，并且 T 满足 $T\leqslant(n-2)$，保证在每次迭代中至少有两个移动是被准许的，那么可以得到下列关系。

- 一个点与其后续沿着轨迹 $T+1$ 步的所有点的汉明距离 H 是严格递增的：

$$H(X^{(t+\tau)},X^{(t)})=\tau，\tau\leqslant T+1$$

- 该轨迹上的最小重复间隔 R 为 $2(T+1)$：

$$X^{(t+R)}=X^{(t)} \;\Rightarrow\; R\geqslant 2(T+1)$$

上述关系清晰地表明**禁忌如何关联到多样化**：T 越大，距离 H 就越大，H 是能够返回已访问的点之前必须经过的轨迹的距离。不过 T 不能太大，否则起始阶段过后，允许移动的个数会大大减少，从而降低了移动的自由度。

上述对禁忌与多样化关系的描述表明，每当字符串中的一个二进制位改变后，它会被冻结 T 个迭代循环。图 22-5 展示了构型 $X^{(t)}$ 的演变过程，要优化的函数是 $f(X) \equiv \text{number}(X)$，$\text{number}(X)$ 将二进制的 X 转化为十进制的整数。禁忌间隔 T 等于 3。

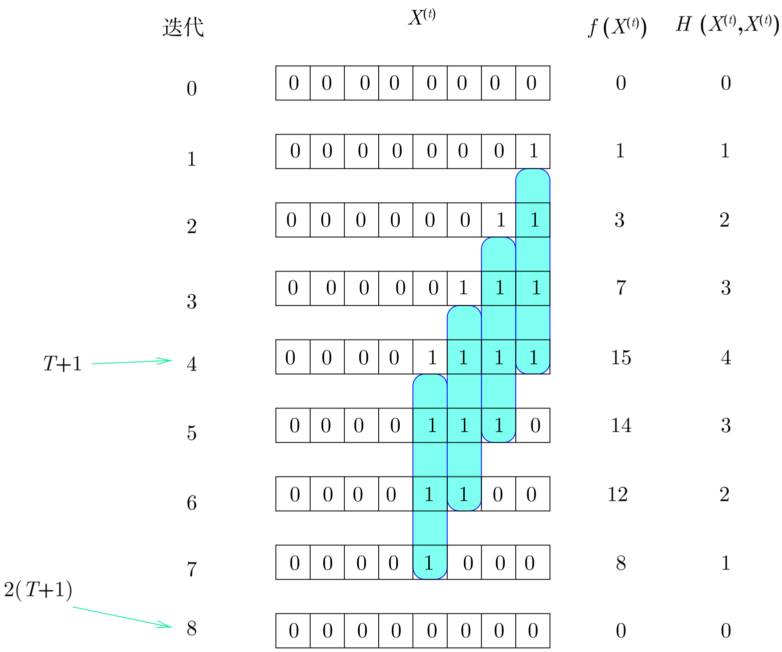


图 22-5 禁忌间隔 T 和多样化之间关联的示意图，多样化通过汉明距离 $H(X^{(t)}, X^{(0)})$ 得到，本例中 $T = 3$。本图改编自参考文献 [18]

以下的物理类比形象地说明了这一现象：当某个二进制位被改变后，就将一个冰块放在它的上面，使得这个二进制位在未来的 T 次迭代中不会再改变。在图 22-5 中，“冻结的”二进制位用阴影框表示，说明该二进制位不能改变。本例中，起始的构型是全为 0 的二进制串，即一个局部最优。在第 0 次迭代，最优的移动改变了最后一位的值。在第 1 次迭代，最后一位的值被冻结了，而且最优的准许的移动改变了倒数第二位的值。二进制串的汉明距离在第 $(T+1)$ 次迭代时达到最大，之后减少，并且第 $2(T+1)$ 次迭代时，起始构型重新出现。当形成一个上述的循环时，从起始构型开始到 $H = T + 1$ 所经过的构型，和从 $H = T + 1$ 开始降低到 0 所经过的构型是不同的。换句话说，这样的轨迹看起来就像是**一个围绕局部极值点的套索**，而且没有人会浪费 CPU 时间去访问已访问过的构型。通常来说，当经过一个局部最优后，可以通过访问其他局部最优得到一个更好的目标值。当然，一旦发现了一个局部最小，所有位于其吸引域（即被局部搜索动力系统收敛到该最小值的所有点）内的点都无须再访问。实际上，通过定义可以知道它们的 f 值大于等于该局部最小。T 值的选择需要保证，当汉明距离达到 $T + 1$ 后，搜索轨迹朝着一个可能存在更优局部最小的新吸引域移动。

因为所需的最小汉明距离（给定吸引域的某种吸引半径）是未知的，所以需要在搜索过程中同时寻找合适的值，通过**反馈的**（reactive）**方式来决定**T。基本的禁忌机制不能保证上述的循环不再发生 [15,16]。此外，对于给定的问题 $(\mathcal{X}, f)$，如果没有关于搜索路径概率上的先验知识的话，T 的选择将是非常困难的。如果搜索空间是非齐次的，T 的大小在某个区域 $\mathcal{X}$ 是合适的，但在另一个区域又可能是不合适的。例如，有时 T 太小，不能避免出现循环；有时 T 太大，只有一小部分移动是被准许的，导致搜索不充分。

RSO 在搜索过程中使用一个简单的机制来自适应地改变 T 值，所以 $T^{(t)}$ 值始终是适合当前问题的局部结构的（因此称作“反馈”）。基本的设计原则是，对于一个给定的局部构型，确定其能逃离某局部最小吸引域的最小禁忌值，如图 22-6 所示。这个基本原则是：T 最初等于 1（仅禁止返回前一个构型）；如果搜索轨迹陷在一个局部最优的吸引域内（不断地访问已访问过的点），T 值将不断增大；如果遇到了从没访问过的区域，这说明进入了另一个局部最优的吸引域，T 值将不断减小。如果该问题只有一个局部最优，即该局部最优也就是全局最优，那么 RSO 的作用就没有那么明显，但是使用也无妨。RSO 将会找出这个局部最优，并储存它，然后尝试去搜索一个更优的点。必须指出的是，现实问题中存在许多局部最优，所以将一个局部搜索方法转化成有效且高效的求解算法这一过程中，RSO 是至关重要的。

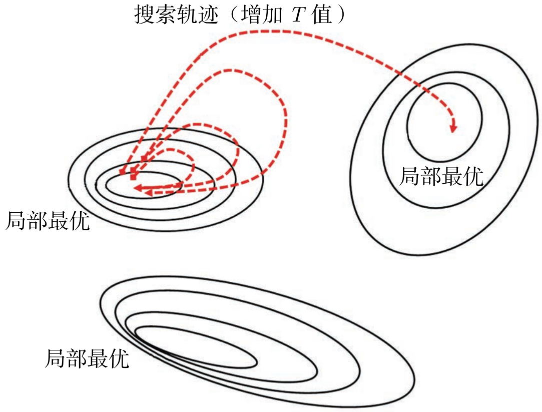


图 22-6 RSO 自适应禁忌。图中描述了 3 个局部最优和优化函数的过程（虚线）。从一个局部最优开始，RSO 在该吸引域内不断搜索，越来越远离该最优点，直到进入另一个吸引域内

RSO 的实现方案需满足搜索过程中的开销（额外的 CPU 时间和内存）限制，即在整个搜索过程中每次迭代的开销基本相等，只需少量的 CPU 周期和内存字节。借助散列函数 ①存

① 散列（hash）是一个有效的方法，也许非计算机领域的大多数研究者并不了解，它为所有数据都建立了字典（连接数据和关键字），所以每次检索数据的时间平均来说是常数级的。散列函数是一个决定性程序，它将任意数据块（这里称作关键字）转换成大小有限的整数，即散列值，所以当数据改变时，其散列值也会改变。一个数据块的散列值可以作为其内存地址，于是数据块的检索是即时的：获得散列值，到相应的内存地址中读取数据。针对不同的关键字共享相同的散列值这一情况，可以使用链表（chaining）来解决，即将包含不同数据块的一个链表关联到内存地址上。

储和检索相关数据，每次迭代的额外内存开销可以减少到几字节，运行时间可以减少到只需从当前构型计算内存地址，并执行少量的变量判断和更新操作 [15]。带禁忌的 RSO 已经被应用于许多问题中，比如组合优化、寻找连续函数的最小值和精粒度亚符号机器学习，部分列表可以在参考文献 [8] 找到。

如果读者对 RSO 感兴趣，可以看看参考文献 [9] 和参考文献 [60] 中提到的关于 RSO 的所有应用实例，或者访问本书在线社区：https://intelligent-optimization.org/。



梗　　概

局部搜索是一种简单有效的方法，能用以确定离散优化问题的改进方案。它生成变化的序列，每一个变化都是*局部的*，即只在有限的部分内影响当前的解决方案。它成功的原因在于，许多问题有着丰富的构型（局部极小值聚在一起，又名**大峡谷**假说），并且相比于对一个全新的解决方案重新进行求值，在当前解决方案的邻域采用**增量求值**会更快。

局部搜索停在**局部最优点**，此时不存在可以改进的邻域，因此当前的搜索轨迹陷在局部最优里。此时需要额外的**多样化的**手段从局部吸引域逃脱。

反馈搜索优化（RSO）在优化的过程中使用学习和自适应，使搜索技术可以根据正在求解的实例和当前暂定解决方案的局部特点进行微调。RSO 可以设计**智能模块来监督基本的局部搜索过程**，又可以平衡多样化和单一化，还可以对优化过程本身的组件（元优化或元启发式）进行优化。

值得注意的是，反馈（reactive）这一术语在本书语境中理解为“对刺激的即时响应”，也可以是积极的“对于未来的问题、需要或者改变的预期做出的行动。”事实上，为了得到反馈算法，设计者需要积极行动起来，通过适当地将模块插入算法，赋予这一算法自主回应的能力。换句话说，**反馈搜索优化算法需要积极的算法设计者**。

第 23 章　合作反馈搜索优化

每当隐修院有重要的事务时，院长要召集全体隐修士向他们说明将要处理的事务。在听取众弟兄的建议之后，他独自考虑，做出他认为最有利的决定。我们之所以主张召开全体会议，是因为天主常常启示年轻人什么是良策。

—— 圣本笃会规，530—550，卡西诺山



如前所述，找到局部最优解的局部搜索（LS），是一种解决复杂离散优化问题的有效构建组件，并且局部极小陷阱可以通过反馈搜索优化解决。本章中将扩展 RSO 以解决连续优化问题，也就是说，通过局部搜索团队来解决输入变量是实数的问题。术语 **CoRSO** 表示，基于战略性地使用记忆以及自适应局部搜索合作团队，来解决连续优化问题的框架。

注意 CoRSO 是一种方法论，而非单一的技术，因此相关的特定技术各自有不同的名字 [17]。CoRSO 的三大组成部分是：主管区域的多重局部搜索器（输入空间的部分）；相互协调；连续“反馈”学习和适应。CoRSO 采用了一个**社会学/政治学上的范式**。

每个局部搜索成员负责一个区域（输入区域），生成样本，并决定何时与其他成员合作。**构型空间的组织化细分**能够适应在线学习方法中问题实例的特征。**通过全局收集的信息进行相互协调**，可以更容易地识别出构型空间中有希望的区域，并分配相应的搜索力度。

23.1 局部搜索过程的智能协作

为了确定符号和搜索方向，假设我们的目标是最小化一个定义在一组连续变量 x 上的函数 $f(x)$。文明进化的模式启发了一套被称为**模因算法**（memetic algorithm，MA）的强大优化技术。根据提出了模因算法的开创性论文[82]，模因算法是基于种群的全局搜索，同时结合了快速的局部启发式搜索，以改进带有创建新个体重组机制的解决方案（甚至达到局部最小）。

改进解决方案的快速启发式搜索，即第 22 章所提到的**局部搜索**（LS）的某种形式。LS 在解空间 $X^{(t)}$ 中产生搜索轨迹，而解空间依迭代计数器 t 的不同而不同，使得下一个点 $X^{(t+1)}$ 从一组邻近点中选择，并且偏向函数值较小的点。正如前文所述，许多实际优化任务的随机局部搜索的有效性，动力源自*邻近点的函数值之间的相关性*。如果某点的值已经很小了，那么在邻近区域找更小值的点就更有可能。

虽然功能强大，但 LS 只能发现*局部最优点*，也不一定是全局最优。该范式是从某个局部最优点（或区域）周围的**吸引域**开始，通过一个离散动力系统在解空间中产生一条轨迹，该轨迹“像流向流域底部的一滴水”。从不同的初始点出发重复运行 LS，可以部分解决局部极小的问题，但这些是完全*无记忆性*（memory-less）的：以前的搜索信息不会影响以后的搜索。

许多情况下，一个给定的优化实例有*不同层次的结构特征*，如第 22 章中图 22-2 所阐释的大峡谷特性。如果将初始搜索空间减少到一组吸引子（局部最小值），可能的情况是，附近的吸引子 —— **吸引域**彼此接近的局部极小值 —— 往往有相关的值。这意味着先前发现的局部最优值的*信息*可以用来指示将来的搜索工作。如果初始点靠近有希望的吸引点，那么更容易发现其他的高质量的局部最优，前提当然是有多元化机制的保障，从而避免回落至以前访问过的点。

在连续串行的局部搜索过程中，积累的有关适应度表面的信息会从过去转移到未来的搜索。然而在并行的局部搜索中，多个局部搜索被激活，信息通过不同的搜索共享部分结果来传递。本书的观点是，区别并不在于串行和并行（可以很容易地使用串行机来模拟并行过程），而在于使用信息的不同方法。这些方法**利用一组局部搜索流积累的信息，为不同的 LS 流进行计算资源的战略配置**。一个 LS 流将被激活、终止，或者更改，这取决于共享的信息库，无论这些信息是在一个中央存储器里，还是以分布式的形式存储（但也进行周期性的信息交换）。

MA 适合出现在这张图中，它是一组由基因所描述的进行遗传进化的个体，探索适应度表面以寻找成功的初始点，而 LS 机制（类似个人的终身学习）使选中的个体充分表达自己的潜力，通过在单个搜索范围中达到局部最优来实现。标准的 MA 所采用的**遗传算法**（GA）遵循选择/复制、交叉和变异的生物学范式。虽然 GA 对于许多问题是有效的，但是对于共享由一组并行的搜索流（又名总体）积累的关于适应度表面的信息而言，根据生物遗传机制的某些特定操作，实际上不一定会优于**人为的直接机制**。毕竟，每个人都希望一生学习到的数学知识可以通过基因传给子孙后代，但生物条件至今未能实现拉马克机制（Lamarckian mechanism）：这一传递由更直接的教育机制来完成。

CoRSO 背后的基本原理是设计某种机制，这种机制具有更高层次的集中协调能力，能有效地管理许多局部搜索流 [17]。我们可以不局限于 GA，随意尝试不同的和更有组织性的方法，来协调一组遵循社会学和政治学范式搜索数据流。不同的搜索过程之间的信息转移，就像高效的政治团体的组织方式，例如僧侣的和谐社区。

23.2 CoRSO：一个政治上的类比

我们喜欢从人类经验中获得的类比，甚于动物或遗传上的类比。政治是一个团体做出集体决定的过程。团体可以是公民政府，也可以是企业、学术和宗教机构等。问题的焦点是商议出一个行动方案来引导决策并得到合理的结果。政治过程旨在制定重要的组织决策，例如各项开支的优先次序，以及基于它们所产生的影响进行选择。

*局部搜索*是一种有效的构建模块，可用于寻找问题实例的初始解，并通过移动到相邻的解逐步构建更好的解决方案。设想一个组织机构，比如公司，公司中的每一员都拥有解决问题的智能。假定每个专家将专业知识用在已制定的初步解决方案上，那么一段时间后应该会拿出各自的改进方案。这样做的目的是从战略上分配工作，以便根据不同专家能力积累的性能数据，来找到优异的解决方案。

经典的模因算法从一个有效的局部搜索开始，借用杂交遗传机制来隐式地积累过去局部搜索的信息，通过传统的生物激励的机制：选择/复制、突变和交叉。第一个发现是，个体能够以更直接和确定的方式利用其最初的遗传信息（初始位置）。这是通过以初始字符串作为*起点*并启动从此处开始的局部搜索，例如搜索局部最优而得到的。术语*模因算法*[73,82] 已被引入某些模型，这些模型包含总体的进化适应，其中个体在生命周期内进行学习。这个术语源自道金斯提出的一个概念，meme，是文明进化的单位，可以表现出局部细化 [40]。实际上，有两个明显的方式可以用来集成个体学习：第一种方式是将局部搜索到的更好解决方案替换初始基因型（*拉马克进化*）；第二种方式通过考虑局部搜索的最终结果而非初始值来修改适应度函数。换言之，适应度函数不评估初始状态，而考虑个体的“学习潜能”值，由局部搜索的结果来衡量。这种进化方式具有改变适应度的效果，同时所得到的进化本质上仍然是达尔文式的。

现在，文化范式的道路已经开通了，自然可以考虑从组织中衍生出的模型，组织中有能够独立学习并进行社会交往的个体，他们也出现在给不同的搜索流进行**战略性的资源分配**当中。尤其是参考文献 [17] 提出了一种混合算法来进行函数的全局优化（称为连续反馈禁忌搜索），其中的快速组合组件（基于禁忌的反馈搜索优化）在一个树状分区的初始搜索空间中确定有希望的区域（盒子），并且在一个有希望的吸引域中随机局部搜索（反馈仿射振荡器算法）找到局部最小值。用人类社会来类比，就像是组织一个大的跨国公司的营销力量（见图 23-1）：每个个体（类比于营销经理）负责某个地理区域，地理区域的大小根据不同地区的利益分配进行调整，分给不同个体的预算与其先前的宣传活动获得的结果相关。另一个政治类比是选

区，根据不同地区的利益（人口密度）进行调整，并按区域潜力来争取资源。

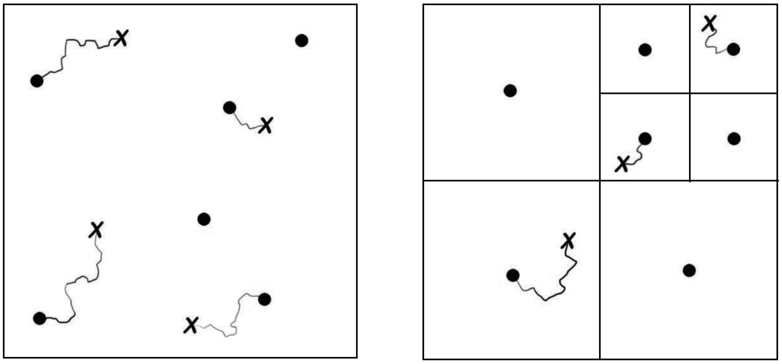


图 23-1 分配局部搜索器的不同方式：模因算法（左图）和政治类比（右图）。叉号表示起始点，圆圈表示使用局部搜索得到的局部最优。在右图中，每个个体负责构型空间的某个区域，比如政治上通过选区来划分区域，企业中为不同的区域分配不同的营销经理。局部搜索流如图所示

不过，现在是时候停止类比，开始考虑具体的算法了。CoRSO 框架的发展遵循下面的设计原则。

- **通用优化**：对需要进行优化的函数 f 没有可微性或连续性的要求。
- **全局优化**：在一个给定的吸引域中，局部搜索部件确定局部最优，组合部件则倾向于在不同吸引域之间跳转，并且偏向那些似乎包含有价值的局部最优的区域。
- **多标度搜索**：在树结构中使用不同标度的网格，在搜索缓慢变化的区域时，能让 CPU 时间有些空闲，并可加强关键区域的搜索。
- **简单性、反馈和适应**：CoRSO 的算法结构简单，该方法的几个参数在搜索期间，通过从记忆中得到的信息自动进行调整。单一化–多样化困境的解决办法是先使用单一化，直到有证据表明需要多样化（当过多区域被搜索轨迹多次重复经过）。如果有证据表明当前区域中包含多个吸引域，那么搜索空间的树形离散化被激活。
- **可调精度**：全局最优值可以用高精度来定位，一是因为网格尺寸的局部调整，二是因为当随机 RAS 收敛时减少其抽样步骤。

在搜索过程中，CoRSO 的特点是能有效地利用记忆，就像反馈搜索优化所倡导的那样。此外，通过在搜索区生成一棵树，简单的自适应（反馈）机制可以用来调整空间的离散化，以及 RSO 作用于禁忌的禁忌期。通过给每个变量设定上界和下界，这种调整限制用户对定义初始搜索区的干预，但不需要调整参数。

CoRSO 框架**将反馈搜索优化与针对特定问题的局部搜索组件融合起来**。最优化问题的一个实例是一对 $(\mathcal{X}, f)$，其中 $\mathcal{X}$ 是一组可行点，f 是需要被最小化的成本函数：$f: \mathcal{X} \to \mathbb{R}$。下面考虑连续优化问题，其中 $\mathcal{X}$ 为 $\mathbb{R}^N$ 的紧子集，由 N 个独立变量 x_i 的界定义，其中

$B_{Li} \leqslant x_i \leqslant B_{Ui}$（$B_L$ 和 B_U 分别是下界和上界）。

在连续优化的许多流行算法中，先从一个起始点开始下降，找到“局部极小”，然后用一个“全局”组件来分散搜索，从而得到全局最优。本书将一个局部最小值 X_l 的相对应的吸引域定义为一个点集，若将其作为局部极小的初始构型，则 X_l 为局部极小值。

某些情况下，正如初始假设所指出的，手头的问题存在一个针对该特定问题的有效局部搜索组件。因此考虑**混合策略**，其中局部极小化的目的是以足够高的精度寻找局部最小，而组合组件的任务是发现有希望的吸引域，使对应的局部极小被激活。因为局部极小化的成本很高，所以只有当一个区域很有可能包含有价值的局部最优时，它才会被激活。与此相反，搜索区域的快速求值通过组合组件执行，并且候选区域的大小要进行调整，使其与单一吸引域的大小相关。当有证据表明至少有两个不同的局部极小都位于同一区域时，这个区域会被分割。

23.3 CoRSO 的例子：RSO 与 RAS 合作

现在简要地给出 CoRSO 框架的一个具体例子，在参考文献 [17] 的原始版本中称为连续反馈禁忌搜索（continuous reactive tabu search）。

作为局部极小化方法，21.5 节描述中的反馈仿射振荡器被用于这种情况，虽然 CoRSO 方法可以与其他局部搜索器合作。在混合方案中，RSO 必须确定能让局部极小化激活的区域。下面来看看这个目标的实现，以及这两个组件的接口。

对于 $i = 1, \cdots, N$, $B_{Li} \leqslant x_i \leqslant B_{Ui}$，初始搜索区域由每个独立变量 x_i 的界所确定，该初始搜索区域被分割的基本结构是一棵包含各区域的树（平行于坐标轴的盒子）。这棵新生的树有 2^N 片大小相等的叶子，即将每个变量的初始值域平均分成两份。只要在该区域中发现两个不同的局部最小值，每个区域就再细分为 2^N 个相等大小的子区域。由于细分过程由 f 的局部特性触发，某些 CoRSO 迭代后，在不同的区域，树会有不同的深度，尺寸较小的区域需要更加专注的搜索。只有树的叶子是组合部件受理的搜索点。树叶划分初始区域：任意两片不同叶子的交集是空集，所有的叶子的并集恰好是初始搜索空间。图 23-2 显示了一个二维任务的典型结构，其中每个叶子区域由实线边框和粗体的二进制字符串标识。

N 维的问题中，每个存在的区域由唯一的 $n \times N$ 位的二进制串 B_S 标识：$B_S = [g_{11}, \cdots, g_{1n}, \cdots, g_{N1}, \cdots, g_{Nn}]$。值 n 是这个区域树中的深度：$n = 0$ 表示根区域，$n = 1$ 表示初始树的叶子（并因此初始字符串有 N 位），当细分一个给定的区域时，n 就增加 1。因此，区域沿着第 i 个坐标的边长，等于 $(B_{Ui} - B_{Li})/2^n$。区域原点位置的第 i 个坐标 B_{Oi} 是

$$B_{Oi} = B_{Li} + (B_{Ui} - B_{Li}) \sum_{j=1}^{n} \frac{g_{ij}}{2^j}$$

对给定区域的邻域求值只针对现有的叶子区域，因为邻域求值的过程并不创造新的分区。

现在，对于一个给定的区域 B，它所对应的标识二进制串为 B_S，在对 B_S 进行基本的操作后，可以在搜索空间里得到 $N \times n$ 个大小相同的区域，图 23-2 展示了 $B_S = (1010, 1011)$ 的情况。因为树在不同的区域可以有不同的深度，所以有些串没有对应的叶子区域，而另一些串可以覆盖多个叶子区域。第一种情况下，对包围其中的最小叶子区域求值；第二种情况下，随机选择一个包围其中的叶子区域进行求值。随机选择在原有区域中以均匀概率产生一个点，并选择包含该点的叶子。这保证了叶子被选择的概率正比于它的体积。

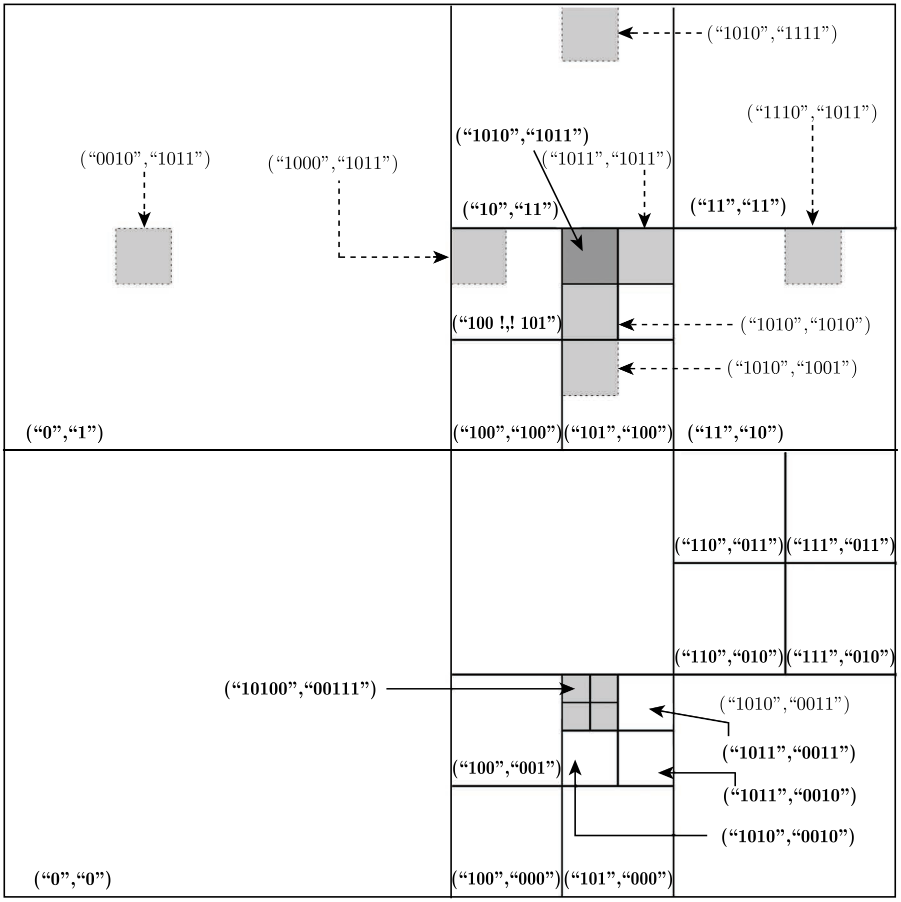


图 23-2 CoRSO：搜索区域的树形结构。实线边框和粗体的二进制串表示叶子区域，阴影表示区域 (1010,1011) 的邻居

1. 不同区域的求值机会

组合优化的 RSO 算法生成由点 $X^{(t)}$ 组成的搜索轨迹，CoRSO 则生成由叶子区域 $B^{(t)}$ 组成的轨迹。此处要强调两个重要的变化：其一，函数 $f(X)$ 必须替换为一个度量，该度量

用来测量当前区域包含好的局部最优的可能性；其二，树是动态的，搜索过程中区域数量会增长。

组合组件必须快速找出有希望的区域。当缺少最小化目标函数 f 的详细模型时，可在区域内以均匀概率分布产生一个点 X，并通过在该点求函数 $f(X)$ 的值来获得 B 区域的简单估计。此处使用相同的函数符号，不同的是，它的参数是显式的：$f(B) \equiv f(\text{rand } X \in B)$。这个估计很简单，其潜在的缺点是，搜索可能会强烈地偏向某个“幸运”区域（例如，$f(X)$ 接近给定区域最小值），或者远离情况相反的区域。为了克服这个缺点，当一个区域在搜索期间再次经过时，生成新的点 X 并求值，返回一些集体的信息。那么，返回的 $f(B)$ 的值则是所求的 X_i 的平均值：$f(B) \equiv (1/N_B)\sum_{i=1}^{N_B} f(X_i)$，其中 N_B 是点的数量。

看一下图 23-2 的例子。当前区域 (1010,1011) 有阴影所示的邻居。左上角的 (0010,1011) 不是一个现有的叶子区域，因此，它被转换成现有叶子区域 (0,1)。反之亦然，(1010,0011) 包含 4 个叶子，其中 (10100,00111) 是一个随机选择的输出。图 23-3 给出了这个例子最终获得的全部邻居。

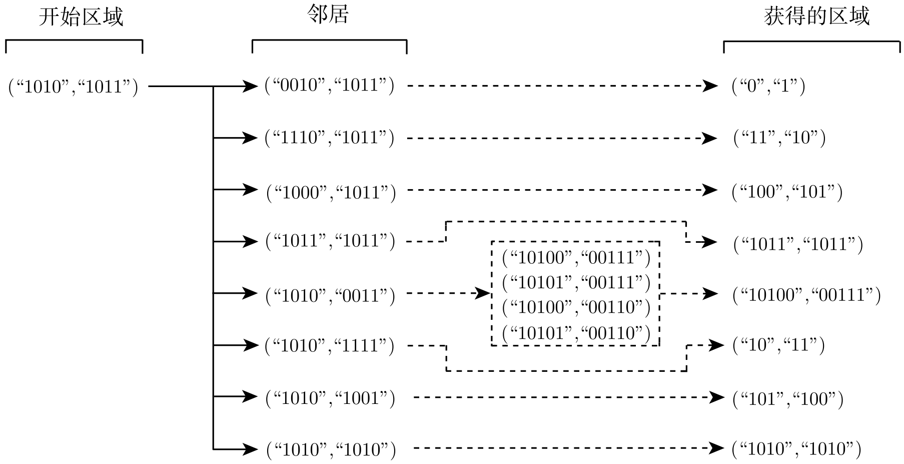


图 23-3　CoRSO：区域 (1010,1011) 所有邻居的值

2．在特定区域激活局部搜索的决定

根据 RSO 动力学，只有当前点对应的基本移动不被禁止时，才对当前区域的邻近区域进行求值。只有当前区域的求值 $f(B^{(t)})$ 小于在附近执行的所有求值时，才决定是否触发局部搜索组件（反馈仿射振荡器）。换言之，激活高精度 —— 因此成本很高 —— 局部搜索的一个必要条件是，相对于候选区域的给定邻域，当前区域有很高的似然性可以产生最好的局部最小值，似然性用 $f(B)$ 测定。由于组合组件的贪婪性，搜索轨迹上的当前区域 $B^{(t)}$ 向着非禁忌局部最优的区域移动，因此它最终将成为局部最优，并满足触发 RAS 的条件。注意，即使一

个给定的区域 B 在这样的竞争中失败（即它不是 RSO 的局部最优），如果搜索期间再次遇到它，它胜出的可能性仍然存在，因为不同的随机点 X 的求值可能产生更好的 $f(B)$ 值。在最佳条件下，如果 f 表面是光滑的，且 $f(B)$ 是一个可靠的指标，并指示着可在区域 B 获得的局部最小值，那么求值方法 CoRSO 会很快。不过该方法在困难的情况下也是具有健壮性的，如果 $f(B)$ 标准差很大或不是可靠的指标，不能清晰地指示通过反馈仿射振荡器获得的好的局部最小值。

当前区域 B 的局部最优是激活振荡器算法的一个必要条件，但并不充分，除非 B 首次达到局部最优。在这种情况下，RAS 总是触发。否则，如果 $r > 1$ 是搜索期间区域 B 被选为局部最优的次数，那么运行另一个 RAS 必须有充分的理由，即找到 B 中一个新的局部最小值的概率足够大。贝叶斯规则用于估计所有局部最优已访问的概率，它可以用于单个小区域，通过从均匀分布的起始点开始重复激活 RAS 来实现多点启动技术。按我们的分割准则，一个给定区域至多有一个局部最优（只要在一个区域中发现两个不同的局部最优就进行拆分，参见 21.4.5 节）。此外，区域的一些部分可以是这样的：RAS 将退出边界，如果初始点属于这些部分。因此，可以将区域划分成 W 个部分，区域中所包含的局部最小值的吸引域和将 RAS 引向外部，并使得吸引域的概率之和为 1（$\sum_{w=1}^{W} P_w = 1$）。

根据参考文献 [26]，若已执行过 $r > W + 1$ 次，且确定 W 个不同的胞格（cell），则“观察区域”的总相对量（即相对体积 Ω 的后验期望值）可由下式估计：

$$E(\Omega|r, W) = \frac{(r-W-1)(r+W)}{r(r-1)}, \quad r > W + 1 \tag{23.1}$$

如果 $r \leqslant W + 1$，那就总是触发反馈仿射振荡器，这是因为上述估计在这种情况下不是有效的；否则 RAS 以概率 $1 - E(\Omega|r, W)$ 再次执行。这种方式中，如果上述估计预测找到一个新的局部最优的概率很小，往往需要增加 RAS 的运行，但是为了健壮性，不会完全禁止重新开始一个运行：式 (23.1) 的贝叶斯估计也许会是不可靠的，或者看不见的部分（$1 - E(\Omega|r, W)$）包含一个很好的最小值和一个小的吸引域。

RAS 的初始条件（如图 21-14 所示）是初始搜索点在 B 内部均匀分布，初始搜索框架是 $\vec{b}_i = \vec{e}_i \times (1/4) \times (B_{Ui} - B_{Li})$，其中 $\vec{e}_i$ 是 $\mathbb{R}^N$ 的标准基向量。反馈仿射振荡器生成一条轨迹，这条轨迹必须包含在区域 B 中，通过将边界区域宽度设为 $(1/2) \times (B_{Ui} - B_{Li})$ 来扩大该区域，并且这条轨迹必须收敛到 B 内的某个点。如果 RAS 退出扩大区或者根区域，它就被终止，通过 RAS 执行的函数求值结果将被丢弃。如果它收敛于原区域之外、扩大区之内，该点的位置将被保存。这两种情况下，CoRSO 组合部件继续照常运行：下一个区域 $B^{(t+1)}$ 是 $B^{(t)}$ 可容邻域里最好的那个。任何情况下，“迄今为止最好的”值总是通过对所有可容的（根区域里面的）点求值来更新。

一个可能的有别于通常 CoRSO 进化的例外情况，仅当该 RAS 在 $B^{(t)}$ 内收敛至一个局部极小值 X_l 时发生。如果 X_l 是找到的第一个局部极小值，它会保存在这个区域的相关记忆

结构中。如果局部极小值的 Y_l 已经存在且和 X_l 对应于同一点，那么它就被丢弃；其他情况下，若树中的“兄弟”（siblings）分开这两个点，则当前区域被分割。分割完成后，当前的区域 $B^{(t)}$ 就不再对应于现有的叶子。为了恢复合法性，在 $B^{(t)}$ 中以均匀分布随机选择一个点，于是合法的 $B^{(t)}$ 变为包含该随机点的叶子区域。因此，初始区域划分中的每个叶子区域被选中的概率都与其体积成正比。分割过程将在下文说明。

3. 根据局部适应度表面调整区域

给定区域 B 内只要确定了两个不同的局部极小值 X_l 和 Y_l，当前区域就被划分为 2^N 个大小相等的盒子。如果 X_l 和 Y_l 分别属于新分区的两个不同的叶子区域，分割就终止；否则，对包含 X_l 和 Y_l 的区域进行继续分割，直到二者分离。

在所有情况下，旧区域不复存在，而被分割所得的集合取代。局部极小值 X_l 和 Y_l 与新划分的盒子关联起来。进行数值计算时，判断两个局部极小值 X_l 和 $Y1$ 为不同的标准是 $\|X_l - Y_l\| < \epsilon$，这里 ϵ 是自定义的精度要求。

CoRSO 终止时，所有已经确定的局部极小值都被保存并报告。

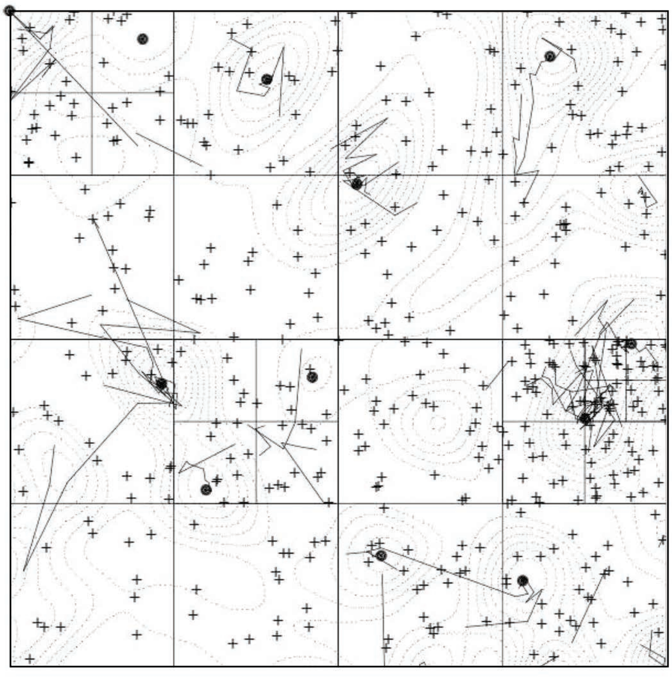


图 23-4 一棵包含适应度平面和求值点的 CoRSO 树：求值点（加号）、局部极值点（实心点）、LS 堆积（折线）。本图改编自参考文献 [17]

图 23-4 展示了一个 CoRSO 运行过程中产生的树状结构的例子，这是一个二维函数的情况（参考文献 [17] 中描述的 Strongin 函数）。局部最优清晰可见，如同“山峰”。注意，求值的

点（用来计算 $f(B)$ 的点）准均匀地（quasi-uniformly）分布于搜索空间：这是与体积成正比进行选择的结果，它保证以公平的方式对待搜索空间的所有区域。RAS 轨迹要么收敛到局部极小值（实心点），要么当其退出扩大区时终止，如 23.3 节所述。按我们的分割标准，每个区域包含至多一个局部极小值。尽管图中没有显示，大多数的点在局部搜索阶段需要求值（本例子中约有 85%），这才是 CoRSO 算法中最昂贵的部分。

需要强调的是，CoRSO 是一种集成局部搜索组件、对区域进行战略分配和大小调整，以及在多维空间产生初始点的方法论。读者可以随意尝试不同的局部搜索组件，改变原空间的分割方法或激发局部搜索的细节。



梗　概

现实世界中很多有趣的优化问题都很复杂，需要大量的时间计算来找出解决方案。

使用多台**计算机并行工作**（也许在云端，必要时可以租用）可以解决这一困难，节省产生可接受的解决方案的时间。

某些情况下还可以考虑**独立搜索流**，周期性地向中央协调器报告目前为止发现的最佳解决方案。永远不要小看这一简单解决方案的力量！其他情况下，各计算机之间更**智能的协调机制**会带来更高的效率。

CoRSO 框架将解空间进行有组织的细分，以在线的方式对其进行调整，以此来协调一组交互的求解器。

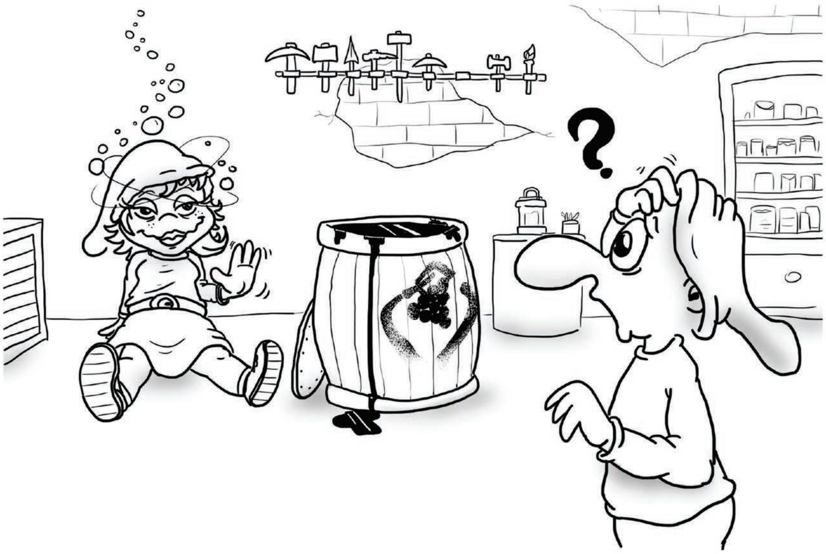
从人类社会组织中提炼出的范式，其特点是能够“边干边学”。相比于从简单的生物或遗传原则派生出的范式，这些范式可能会得到更好的结果。

明智的人能比病毒更有效地解决复杂问题，通常也不会像病毒那样造成致命的后果。苍蝇的一生很短暂，也学不了很多东西，如果一不小心碰到滚烫的白炽灯灯泡，很容易就会送命。人类的孩子则只要触摸一次发烫的灯泡，知道什么是“烫”以后，将来就不敢再这样做了。

第 24 章　多目标反馈搜索优化

好酒若吝惜，怎得佳人醉。①

—— 意大利谚语



生活是充满妥协的，正如民间智慧所言，人们不能“用一个屁股骑两匹马”。大多数现实世界的问题，无法找到一个简单的或者在数学上十分清晰的函数 $f(x)$ 来进行最小化。

有两个重要的难点。第一点，大多数的问题需要达成**一个以上的目标**。这是**多目标优化问题**（MOOP），因此要在许多相互冲突的目标中进行权衡。大多数现实问题都是这样的。例如，你买车的时候，心里有不同的目标，速度、成本、大小等，并且你有自己权衡不同目标的方式。如果你买了一辆法拉利，你的权衡方式很可能不同于那些买了城市小型车的人。如果你正在寻找一个伴侣，不同的候选人可能有着美丽与智慧的不同组合（先接受它，这只是一个粗略的简化）。不幸的是，同时最大化这两个参数的情况是罕见的，因此不得不妥协。

第二点，即使存在一个要被最大化的整体效用函数（cutility function），它抽象地组合两个或更多的目标，然而**得到一个封闭形式的函数却可能是非常困难的**，甚至是不可能的。试着问问你最好的朋友（最好不要直接问你的伴侣）：“跟我说说，你认为美丽与智慧的最优结合

① 意思是鱼与熊掌不可兼得。—— 译者注

的效用函数是什么？”或者，把模型限制为线性组合：“能不能告诉我，美丽和智慧的权重是什么呢？”问题的求解及优化技术往往提供了大量的潜在解决方案，例如设计过程创新、虚拟原型设计、业务流程设计。决策者要从大量的潜在解决方案中选择一个优选的解决方案，这一关键任务需要考虑显式定义的目标（最大化一个或多个数学函数），硬约束和软约束，以及**那些隐式的但往往对于智能决策而言至关重要的偏好**。

问题求解通常是**带有学习的迭代过程**，如图 24-1 所示。其中有一条两个实体之间的学习路径：决策者和配套的软件系统。决策者分析一些代表性的解决方案，了解具体的可能性，并更新目标。软件系统则记住用户的偏好，并将关注的焦点转移到最终用户更为关心的决策/解空间，以此来修改内部搜索过程。该迭代过程持续进行，直到发现一个令人满意的解决方案，或者失去了耐心。

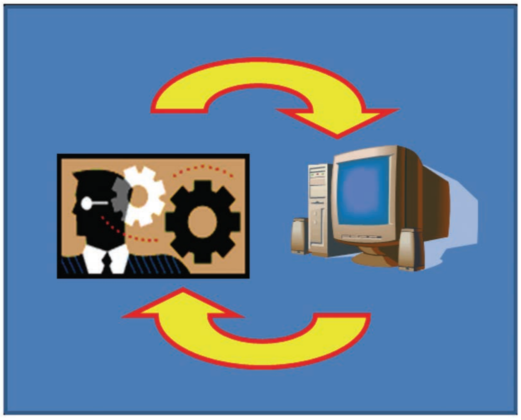


图 24-1　问题求解和优化通常是带有学习的迭代过程

MORSO（多目标反馈搜索优化）用来表示多目标优化任务求解的方法，其特点是迭代和学习路径。学习发生在用户的脑中，以及求解技术（以及相应的软件工具）中。与此密切相关的一个术语是交互式多目标优化，但本书意图以更直接的方式强调系统化、自动化和在线学习技术。

24.1　多目标优化和帕累托最优

在多目标优化问题（MOOP）的典型情况中，用户能够指定一组期望的目标，但并不能给出权衡的方式、不同目标的相对重要性，以及如何将其适当地组合成一个整体的效用函数。一个 MOOP 可以表述为：

$$\begin{array}{ll}\text{最小化} & \boldsymbol{f}(\boldsymbol{x})=\{f_1(\boldsymbol{x}),\cdots,f_m(\boldsymbol{x})\} \\ \text{使服从} & \boldsymbol{x}\in\Omega\end{array}$$

其中 $\boldsymbol{x} \in \mathbb{R}^n$ 为包含 n 个决策变量的向量；$\Omega \subset \mathbb{R}^n$ 是可行域，并且通常由一组决策变量上的约束指定。在之前寻找合适伴侣的例子中，Ω 可以设为某一性别（男或女）的所有人的集合，并且至少有读写能力。你当然不会从不满足这些约束的人中选择伴侣。

向量 $\boldsymbol{f}: \Omega \to \mathbb{R}^m$ 是 m 个目标函数，需要同时加以最小化 ①。目标向量是决策向量的像（image），可以写为 $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x}) = \{f_1(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x})\}$。如果目标函数是相互矛盾的，那么上述问题就是不适定的，这种情况在现实世界里经常发生。这些情况下，如果一个目标向量的任何一个分量的改进都不得不损害其他分量，那么这个目标向量就被认为是最优的。如果对于所有 k 都有 $z_k \leqslant z'_k$，并且存在至少一个 h 使得 $z_h < z'_h$，那么称目标向量 $\boldsymbol{z}$ 对 $\boldsymbol{z}'$ 占优，表示为 $\boldsymbol{z} \prec \boldsymbol{z}'$。如果对于某个点 $\hat{\boldsymbol{x}}$，没有其他点 $\boldsymbol{x} \in \Omega$ 使得 $\boldsymbol{f}(\boldsymbol{x})$ 对 $\boldsymbol{f}(\hat{\boldsymbol{x}})$ 占优，那么称点 $\hat{\boldsymbol{x}}$ 为**帕累托最优**（Pareto-optimal）。图 24-2 展示了这个概念。**帕累托边界**（或帕累托前沿）包含了所有帕累托最优的点。在这个例子中，一个伴侣是帕累托最优的，如果找不到同样好看但更聪明，或者同样聪明但更好看的人，等等。就像你意识到的那样，只考虑帕累托最优人选是有道理的：没有哪个理性的人会喜欢一个不占优的伴侣！通过将注意力限制在帕累托边界（帕累托有效的那一组），设计者可以在此组内做出权衡，而不是考虑每一个参数的全部可能性。

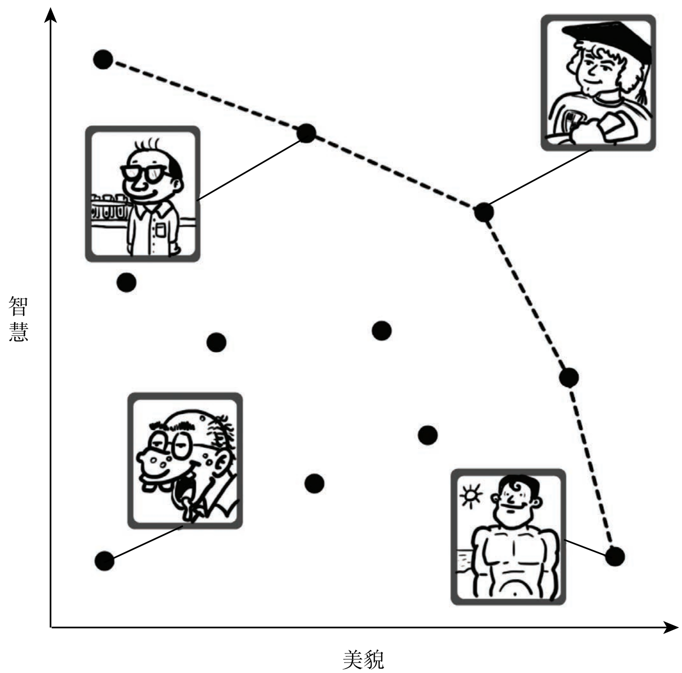


图 24-2　帕累托最优。所有像左下角那个人一样不占优的点都不会被列为最终候选。在帕累托边界（虚线）上的点需要进行权衡

① 如果目标函数是美貌和智慧，那么显然必须加以最大化，即最小化其反面。

正如前文所提到的，让用户在看到实际优化结果之前*先验地*量化效用函数（例如通过选择不同目标的线性组合权重）是具有挑战性的。如果最终用户与优化专家合作，他们之间可能出现误解。当 MOOP 的冲突目标数目增加时，这个问题可能变得更严重。最终用户可能不满意优化专家提出的解决方案，因为一些目标始终隐藏在最终用户的脑海中。*后验*方法也存在不同的缺点，该方法给最终用户提供一组代表整个帕累托前沿的解决方案，让他可以从中选择最喜欢的解决方案。

虽然对用户来说，提供明确的权重和数学公式是很困难的，但是他们一定可以*评估返回的解决方案*。大多数情况下，改进这种状况的策略是**最终用户和优化专家互动合作来改变问题本身的定义**。优化工具将在新版本的问题上重新执行。这个过程可以被迭代任意次数，如图 24-1 所示。

24.2 脑–计算机优化：循环中的用户

现在我们正进入本书最前沿、最令人兴奋的话题：分析、可视化和优化的整合。抽象解决方案都是有特定**含义**的数字向量。

在**互动问题求解**的场景中，用户可以调用一个针对特定问题的程序来获得有关特定解决方案的信息。该程序是各种不同的应用都要提供的，可用于可视化特定点的详细信息，例如解决方案的图形显示。

针对特定问题的同一程序可以用于接受特定解决方案的**反馈**，比如个人的评价，如图 24-3 所示。

作为一个经验法则，大部分为解决现实世界问题所做的努力，都花费在**问题的定义上**，以一个可计算的方式指定需要优化的函数。在此建模工作完成后，优化在某些情况下成为商品。给研究人员和开发人员的提示是，更应该致力于设计配套的技术和工具来帮助最终用户，他们往往没有数学和优化的专业知识来定义和完善需要优化的函数，使其对应于真实目标。想想寻找一个伴侣时，定义自己喜欢的美貌与智慧的权重。如果有人要你在开始搜索前就以定量的方式来指定权衡，你可能会感到很尴尬。只有在看到一些例子之后，你才能弄清权重和目标。

反馈搜索优化（RSO）致力于在线学习技术，以支持通过**自适应局部搜索方法寻求解决方案，这一方法与搜索历史相关**。学习的信号包括在该实例上运行算法时收集到的结构特征数据，例如吸引域的大小、轨迹中的陷阱、以前重复访问过的解。该算法通过与一个先前未知的环境交互进行学习，这一环境由现有的（固定的）问题定义给出。

我们认为还有一个有趣的在线学习循环，在这个循环中，学习的信号来自最终用户，旨在**修改和细化问题定义**本身。这种情况可能发生在许多环境中，取决于给定的关于问题的先验知识的多少、允许的修改，以及问题的种类。

参考文献 [10] 中描述了使用 RSO 的交互式多目标优化的一个例子。本书考虑的方法和

参考文献 [121, 41, 103] 有共同之处，通过对解决方案进行两两比较实现了与最终用户的交互，但使用非线性偏好函数来解决更广泛的一类问题。这种情况是有趣的，因为许多（也许是大多数）决策问题都是非线性的，这反映了我们的偏好，即合理的、折中的解决方案。对于任意（非线性）模型这一领域的前沿技术感兴趣的读者可以读读最近的技术文章 [13]。

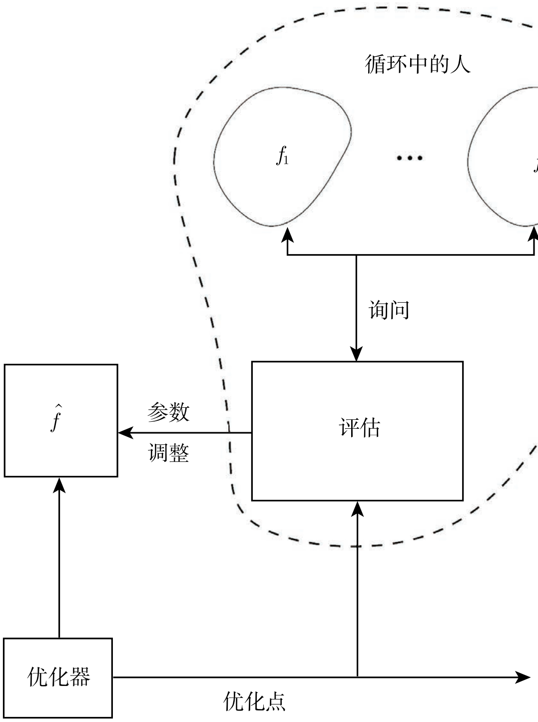


图 24-3　脑–计算机优化（BCO）：对于交互的多目标优化，从最终用户中学习问题的定义（改编自参考文献 [10]）。

这里我们关注简单和经典的线性情况 [10]，目标是**学习最终用户偏好的占优的解**。假设用户为 MOOP 问题提供了不同的目标，然而看到优化的实际结果之前，他不能量化不同目标的**权重**。该系统的目的是学习权向量 $\boldsymbol{w}=(w_1, w_2, \cdots, w_m)$，它最优化如下线性组合 g：

$$g(\boldsymbol{x}, \boldsymbol{w}) = w_1 f_1(\boldsymbol{x}) + w_2 f_2(\boldsymbol{x}) + \cdots + w_m f_m(\boldsymbol{x})$$

一种更紧致的形式是：

$$g(x_1, x_2, \cdots, x_n, \boldsymbol{w}) = \boldsymbol{f}(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{w}$$

其中 $\boldsymbol{f}=(f_1, f_2, \cdots, f_m)$。不失一般性的条件下，假设 g 必须被最小化。

每次迭代提出两个解决方案，决策者从中选择最喜欢的那个，如果有的话。这是要问的最简单的问题，一个定性的整体偏好。如果决策者不能回答，他或许应该换一份工作！由最终用户说明的偏好转换成权重必须满足的约束。这保证所得到的效用函数与用户的判断是一致的。如果 $\boldsymbol{a}=(a_1,a_2,\cdots,a_n)$ 和 $\boldsymbol{b}=(b_1,b_2,\cdots,b_n)$ 是系统提供的两个解决方案，最终用户的偏好是 $\boldsymbol{a}\prec\boldsymbol{b}$，即相比于解决方案 b，用户更偏好 a ，可通过以下约束来表示：

$$g(\boldsymbol{a},\boldsymbol{w})<g(\boldsymbol{b},\boldsymbol{w})$$

因此，最终用户回答的每个问题都会产生新的权重上的线性约束。学习用户偏好的问题就变成了寻找一个解 $\boldsymbol{w}$，满足由用户反馈所产生的权重约束集。

所有的权重都以位于区间 $(0,1)$ 的随机值初始化，然后归一化，使其和为 1。在每次迭代中，两个占优解 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 由最终用户进行比较。这两个解都是通过最小化输入问题的目标函数的线性组合而得到的：

$$\min_{\boldsymbol{x}} g(\boldsymbol{x},\boldsymbol{w})$$

特别是第一个解 a，是使用当前的权向量 $\boldsymbol{w}^{\text{curr}}$ 得到的，该向量通过*最中间权重*（middlemost weight）技术 [86]，求解下面的线性规划问题而得到：

$$\begin{aligned}&\max_{\boldsymbol{w}} \quad \gamma\\ &\text{使服从}\quad\begin{cases} g(\boldsymbol{a},\boldsymbol{w})\leqslant g(\boldsymbol{b},\boldsymbol{w})-\gamma & \forall \boldsymbol{a}\prec\boldsymbol{b}\\ w_i\geqslant\gamma & \forall i=1,\cdots,m\\ \gamma\geqslant 0 \end{cases}\end{aligned}$$

上式的意义是搜索到的权重是*一致的*，但也要*远离*一致区域的*边界*。正参数 γ 越大，不等式就越安全。即使添加受限噪声（例如物理量的测量误差引起的），且 g 值稍有变化，不等式方向改变之前仍有一个*安全间隔*（safety margin）。

第二个解 $\boldsymbol{b}$ 可以使用权向量 $\boldsymbol{w}^{\text{pert}}$ 得到，该向量通过扰动 $\boldsymbol{w}^{\text{curr}}$ 生成，并确保产生的两个解足够不同。参考文献 [10] 中考虑了不可行线性约束问题（可能由于决策者一时糊涂，或者线性逼近太粗糙）。更复杂的非线性情况的解决办法，是基于参考文献 [13] 中支持向量机方法的机器学习技术。

参考文献 [33] 提出了帕累托前沿的主动学习（ALP）的新方法。ALP 将标识帕累托前沿转变成一个有监督的机器学习任务。产生监督信息的计算工作量由一个主动学习战略来降低。值得一提的是，该模型是从一组有信息量的训练目标向量中训练得到的。

本章最后想说的是，请记住，如果你需要解决一个有挑战性的问题，*智能优化的力量之源既能帮助定义你想完成的任务，又能实际计算出一个或多个解决方案*。

许多情况下，一些决策可能，并且也许应该推迟，直到有专家用户对初步可能的解决方案做出评价。



梗　概

当顾问访问一个企业时，如果以传统的数学方式来优化，那么他会问的典型问题是：“您的企业中，要优化的函数是什么？”他说“函数”的意思是明确的数学模型，将输入（决策）和输出（如利润）联系起来的公式，没有任何含糊之处。这种态度再加上大多数企业缺乏明确定义的模型，也许就是把最优化的能量扼杀在现实世界中的原因。

企业所有者告诉顾问：“对不起，我没有数学函数。”**LION 方式**打开了一扇希望之窗，并带来了释放优化能量的机会。他可以回答说：“不要担心，即使您不能给我您的模型，我可以**根据您的数据和反馈**建立一个模型给您。”使用一台个人电脑来支持决策，并不会让你废弃个人的专业大脑。

大多数问题求解和优化工作本质上是**有学习参与的迭代过程**，从数据中学习，也从决策者那里学习。当这一做法得到公认，我们会迎来一个充满机会的崭新时代。现在仍然需要很多的努力，对数据科学家来说是个好消息，不过前进的道路已经在地图上了。

第四部分

应 用 精 选

第 25 章　文本和网页挖掘

百科全书将出现全新形式，它会是一张有相互关联的条目贯穿其中的网，被当作记忆的扩展存储器（memex）并被放大。

—— 范内瓦　布什，1945



机器学习和优化的应用是不计其数的。接下来的章节考虑了两个例子：文本挖掘，它本身就是一个完整的领域；合作推荐，企业从简单客户数据中提取有价值的信息的一个典范案例。

如果数据是非数字元素的集合，例如文档，我们仍然可以使用很多分析数字数据的技术，但是需要**对文档进行适当的预处理，并对 ML 方法进行微调**。预处理将文本转换成包含数值的向量。ML 的微调方法要处理很多情况：向量的坐标数可能十分庞大，文字具有歧义，文本结构难以分析，还有需要特设度量、特征选择和提取的情况。

信息检索大多用于搜索文档和文档中的信息。**网页挖掘**则是调整方法以适应于万维网的情况。网页是一种非结构化（或者最多半结构化）数据集，主要是人类可读的文本和图像的形式，通过超链接相连。网页不是一个数据库：数据项结构（模式）的完整描述是没有的，它只是人类可读的数据和可用的超链接的一个混乱集合。已经有一些工作帮助机器（计算机）通过语义支持来自动提取网页的意思，但是由于其无组织性和不断发展的结构，这一任务十分艰巨。“语义”指数据项所表达的“意思”，所谓的语义网（Semantic Web）就是向网页添加元

数据 —— 关于数据含义的数据 —— 使得自动代理和软件能够更智能地访问网页，例如理解某个字段是人名，某个字段是年龄，某个字段是地址等。

“大数据”是一种流行的商业术语，它所描述的数据集合是如此之庞大、复杂和非结构化，以至于传统数据处理的应用程序难以应付。

记住，网页所包含的除了文本，还有标签（tag），它们修改外观和文本的含义，其中包含到其他文档的链接（**超链接**），一些情况下还有**元数据**描述不同部分的含义。图 25-1 展示了一个简短的 HTML 网页的例子。*没有网页是一座孤岛*：其实超链接对网页的搜索、排名、分类都有帮助。当然，其他领域也存在类似的链接结构，如社交网络和论文参考文献等。

```
<html>
 <head>
  <title>Learning and Intelligent Optimization</title>
  <meta name="author" content="Roberto Battiti">
  <meta name="keywords" content="LION, ML, optimization, big data">
 </head>
 <body>
  <h1>The LION way is the future</h1>
  The reasons are explained in the
  <a href="intelligent-optimization.org"> LIONlab homepage </a>.
 </body>
</html>
```

图 25-1　一个超文本标记语言（HTML）网页的例子，HTML 是网页的标准语言，描述了整个页面的结构

25.1 网页信息检索与组织

在冒险尝试更有趣的网页挖掘任务之前，如排名、聚类和分类，先开始了解如何收集网页的原始内容（**爬虫**），以及如何组织这些内容为进一步的分析做准备（**索引**）。如果你不关心原始数据是如何得到的，只对高层次的网页挖掘问题有兴趣，你可以直接跳到 25.2 节。

所收集的文档经过处理后，成为适于回答查询和检索信息的**索引**。不同于 RDBMS（关系数据库）的情况，*回答的顺序是基本的*：用户希望首先看到相关数据。换句话说，目的是最大化前几个答案就能满足用户需求的概率。网络爬虫和网页索引的结合就是一个**搜索引擎**。

在某些情况下，为简化搜索而建立**目录**。它们是树形结构的（分类），最初是人工设计的。文件的组织过程可以自动通过**聚类和无监督学习**方法来完成。其目的是自动发现文档的分组，使得同一组内的文档比不同组中的文档*更为相似*。正如人们所认为的，设计自动化的文档聚类技术时，相似性度量是一个至关重要的问题。

25.1.1 爬虫

网页信息的处理始于**爬虫**，一个访问网页和收获其中所载信息的系统方法。爬虫的基本原理包括从一组给定的 URL 开始访问网络，获取和收集相应的页面，扫描收集到的网页来找

出未被收集的网页超链接。

如果你熟悉图论，用节点表示网页，边表示链接，任务就是*遍历图*，即以系统化的方式访问所有节点，同时避免重复访问。基本爬虫对网络的访问能够与底层通信协议（HTTP）的一些知识一起使用，虽然如此，由于要避免很多陷阱，仍需仔细考虑设计。

- 许多网页服务器假设人的意愿是网页请求的动力，因此，它们认为任何每秒获取许多网页的尝试都是攻击，并以拒绝访问来响应。
- 在互联网上，越来越多的网页在细节上是动态的，其内容取决于用户预先输入的数据，也取决于预先存在客户端的 cookie，甚至还取决于请求来自的位置；因此，任何自动收集所有可用信息的企图都会失败，系统事先必须获取一些用户的信息。
- 解析主机名可能比获取数据本身需要更长的时间；一般情况下，确定真正的瓶颈并不是件容易的事。
- 现在网页中占主导地位的是带有多对多关系的虚拟服务器（域名对应 IP 地址，URL 对应网页，何况还有镜像或抄袭的信息），于是那些已经被访问的网页越来越难以识别。

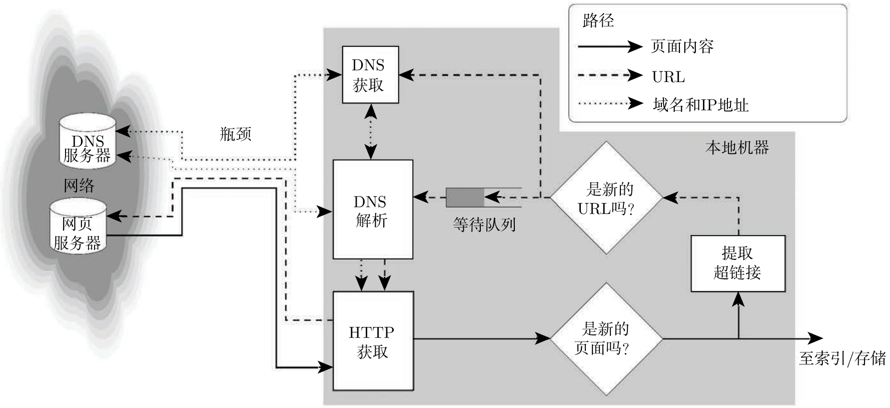


图 25-2　一个基本的爬虫结构：从网页中提取 URL 并加入队列；提前获取域名解析以防潜在的 DNS 瓶颈

图 25-2 展示了一个极简的爬虫结构：尚未访问过的网址保存在一个队列中；每获取一个页面，扫描该页面查找新的 URL。为了克服上述的 DNS 瓶颈，可以在该 URL 的最终请求之前发出一个初步的 DNS 请求。避免页面和 URL 重复也很重要，因此可以在整个工作流程中放置各种“是新的吗？”检查点。

25.1.2　索引

索引是必需的**预处理，使查询可以迅速得到回答**。这是最简单并且迄今为止最常用的一种查询，包括一个或一个以上的条件，在某些情况下，这些条件由布尔运算符相连接。例如，

人们可能搜索含有单词“Reactive”，而不含单词“Search”的文档；包含短语“Reactive Search Optimization”的文档；“Reactive”和“Search”出现在同一个句子里的文档，等等。

建立索引之前，文档先进行一系列清洗步骤，通常包括：HTML 标签和其他非相关的标记项被过滤掉（有一些例外 —— 应该保留一些元信息，标题标签可能会提供相关性的信息和文字的可见性）；标点符号可以移除，如果需要也可以更换为句尾标记；字母大小写统一（例如全部小写）；其余文本都标记化（tokenized），即将其分为单词；很常见的词语（“and”“I”“the”，等等），也被称为停止词（stopword），需要被除去；同一个词的不同形式都转变为词干（这样“play”“playing”和“played”都对应同样的标记）。

虽然在这个过程中，并非所有的原始信息都被保存，从信息检索的角度来说，丢失的部分主要是噪声，如果用户发送查询“Shakespeare play”给搜索引擎，他会期望含有“Shakespeare plays”的结果，并具有合适的大小写和单复数形式。

图 25-3 展示了两个样本文档[①]，d_1 和 d_2，其中下标表示符号在文档中的位置。

直接索引是将词项 ID `tid` 映射为文档中的 ID 和位置 (`did`,`pos`) 的一张表。图 25-3 的左侧展示了这样的一张表，在这张表里搜索包含某符号的所有文档是非常低效的（必须扫描整张表）。反向索引通过“转置”前一张表得到（图 25-3 的右表），并且给出对应每个符号的包含其在内的文档列表。

$d_1 = \text{My}_1\ \text{care}_2\ \text{is}_3\ \text{loss}_4\ \text{of}_5\ \text{care}_6,\ \text{by}_7\ \text{old}_8\ \text{care}_9\ \text{done}_{10}.$

$d_2 = \text{Your}_1\ \text{care}_2\ \text{is}_3\ \text{gain}_4\ \text{of}_5\ \text{care}_6,\ \text{by}_7\ \text{new}_8\ \text{care}_9\ \text{won}_{10}.$

tid	did	pos
my	1	1
care	1	2
is	1	3
⋮	⋮	⋮
new	2	8
care	2	9
won	2	10

tid	pos list
my	$d_1/1$
care	$d_1/2,6,9\ //\ d_2/2,6,9$
is	$d_1/3\ //\ d_2/3$
loss	$d_1/4$
of	$d_1/5\ //\ d_2/5$
by	$d_1/7\ //\ d_2/7$
old	$d_1/8$
done	$d_1/10$
your	$d_2/1$
gain	$d_2/4$
new	$d_2/8$
won	$d_2/10$

图 25-3　两个文档（顶端）及其直接索引（左表）和反向索引（右表），来自参考文献 [34]

① 选自莎士比亚的《理查二世》，第四幕，第一场。

25.2　信息检索与排名

网页的原始内容被正确保存和预处理（添加索引）之后，现在来考虑更有趣的任务，搜索文档和文档内包含的信息，也称为信息检索（IR）。在一般情况下，人们想检索到与查询相关的、质量良好的文档。如果搜索“loss”和“care”，可能检索到莎士比亚，但也可能是“hair loss care”相关的文档，这很可能是你不想要的结果，如果你的兴趣是文学，而不是防脱发护理的话。

文档检索性能指标的标准定义如下。如果 A 是相关的文档，B 是检索到的文档（见图 25-4 以及 3.3 节的图 3-4），可以确定：

- 检索到的相关文档（真阳性）为 $A \cap B$；
- 检索到的无关文档（假阳性）为 $B \setminus A$；
- 未检索到的相关文档（假阴性）为 $A \setminus B$。

检索系统的**精确率**（precision）定义为检索到的文档中相关文档的比例：

$$\text{精确率} = \frac{|A \cap B|}{|B|}$$

该系统的**召回率**（recall）定义为相关文档被检索到的比例：

$$\text{召回率} = \frac{|A \cap B|}{|A|}$$

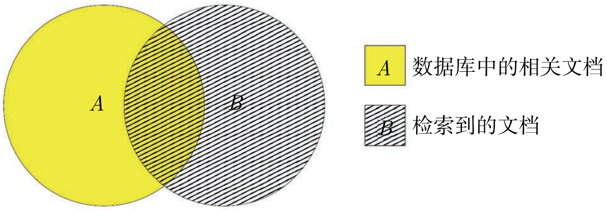


图 25-4　信息检索：相关文档和检索到的文档

召回率度量与网络搜索通常不那么相关，因为相关文档的数量对于人工检查来说通常是过大的。而对于搜索引擎而言，呈现给用户的结果的次序是至关重要的。一般情况下，意味着对文档进行排名，因此适当的性能指标应该青睐那些将相关文档放在最前面的方法，并在用户浏览器中首先显示它们，作为对搜索的响应。

考虑图 25-5，其中浅色点代表相关文档，右侧显示了排名。很明显，最优排名过程应该将浅色文档放置在顶部。考虑到这一点，下面来介绍性能的更加专业的定义。

设 D 是包含 $n = |D|$ 个的文档的语料库，令 q 是一个查询。$D_q \subset D$ 定义为与 q 相关的所有文档。假设 D_q 表示该系统“所希望”的答案。令 $(d_1^q, d_2^q, \cdots, d_n^q)$ 为 D 的一个次序（“排

名”），是系统对查询 q 返回的响应。$(r_1^q, r_2^q, \cdots, r_n^q)$ 定义为

$$r_i^q = \begin{cases} 1 & d_i^q \in D_q \\ 0 & \text{其他情况} \end{cases}$$

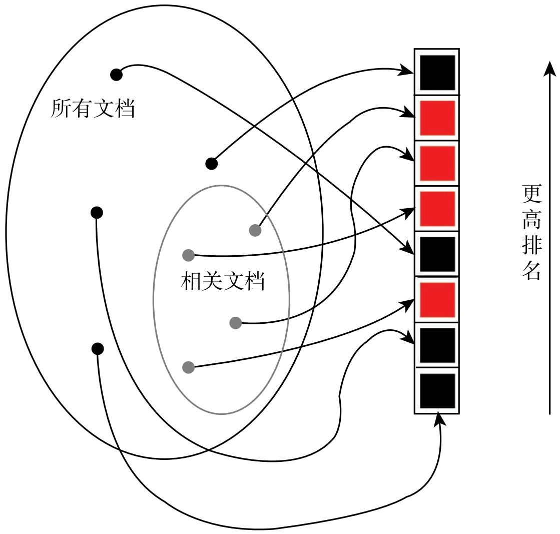


图 25-5 一个排名的例子

现在可以定义排名相关的召回率和精确率，这将有助于回答这个问题：“如果只用了排在前 k 名的答案，我们如何评价系统性能？”

排名为 k 的召回率定义为在前 k 个位置找到相关文档的比例：

$$\text{召回率}_q(k) = \frac{1}{|D_q|} \sum_{i=1}^{k} r_i^q$$

类似地，精确率定义如下：

$$\text{精确率}_q(k) = \frac{1}{k} \sum_{i=1}^{k} r_i^q$$

与往常一样，没有免费的午餐：在分析排名列表时，召回率可以通过增加 k 的值来提高；但如果这样，就会出现越来越多的不相关文档，从而拉低了精确率（**精确率–召回率平衡**）。

25.2.1 从文档到向量：向量–空间模型

为了使用向量空间的标准技术来搜索、聚类和分类文档，首先需要将每个文档映射到一个向量（**向量–空间模型**）。

预处理后，现在你的文档是单词的一个包，实际上是标记的一个包，而得到一个向量的最简单方法是：i) 固定一套单词（标记）；ii) 每个单词（标记）都由一个单独的数轴来表示；iii) 如果文档不包含某符号 t，那么向量对应数轴 t 上的值设为零；如果在文档中包含该词一次或更多次，那么设为一个大于零的数。

如果 $n(d,t)$ 是文档 d 包含单词 t 的次数，那么单词 t 在文档 d 中的**词频**（term frequency）$\mathrm{TF}(d,t)$ 定义为关于 t 在 d 中的相对频率单调递增的数字。一些可能的定义如下：

$$\mathrm{TF}(d,t) = \frac{n(d,t)}{\sum_\tau n(d,\tau)}$$

$$\mathrm{TF}_{\mathrm{SMART}}(d,t) = \begin{cases} 0 & n(d,t)=0 \\ 1+\log\left(1+\log n(d,t)\right) & \text{其他情况} \end{cases}$$

如果一个词出现的次数太多，可用该 $\mathrm{TF}_{\mathrm{SMART}}$ 式子避免某一维上出现夸张的值。在网页出现的最初几年里，这是一个常见的情况，当时简单的搜索引擎只计算词出现的次数。曾经有很多网页为了在用户搜索时排在前面，故意包含像“sex”这样的词，重复出现多达数百次。事实上，最近的搜索引擎通过使用超链接信息来打击这种垃圾信息，后面的章节将会提到。

事实上，最有趣的词往往是文档中不太常见的词（**罕见词**如“C++”“反馈搜索优化”“随机的”，它们可能比“是”“好”“自由”“优秀”包含更多的信息）。**逆文档频率**可以定义为一个随着某个词的整体词频在全体文档语料库中增加而单调递减的数字：

$$\mathrm{IDF}(t) = \log\frac{1+|D|}{|D_t|}$$

其中 D_t 是包含词 t 的文档集合，对数用于避免那些非常罕见的词带来的夸张乘数。值得注意的是，上述方法得出的向量是启发式的，并不是基于信息理论那样的基本原则。如果你认为对数用在这里不合适，可以随意尝试其他的函数。降低了出现在太多文档中的弱词的重要性后，TF-IDF 空间（词频逆文档频率）中的特定文档 d 由下面的向量表示：

$$\boldsymbol{d} = (d_t)_{t\in\mathrm{terms}} \in \mathbb{R}^{\mathrm{terms}}$$

其中分量 d_t 是

$$d_t = \mathrm{TF}(d,t)\mathrm{IDF}(t)$$

查询 q 是由词组成的一个序列，因此允许其表示 $\boldsymbol{q}=(q_t)$ 和文档位于同一空间。给定查询 $\boldsymbol{q}$ 和文档 $\boldsymbol{d}$，现在通过考虑向量空间相似性的度量，可以测量它们的邻近度，如图 25-6 所示。下面列出了两个在 TF-IDF 空间中经常使用的相似性度量。

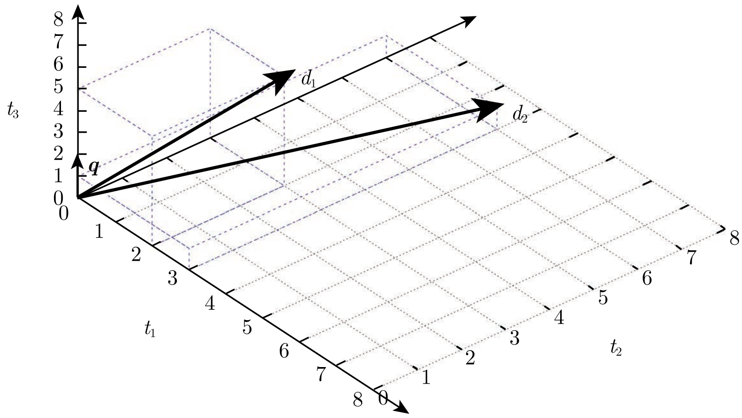


图 25-6 几何解释

- 欧氏距离 $\|\boldsymbol{d}-\boldsymbol{q}\|$。为了避免量极的不统一，向量应被归一化，也就是说，原文档 $\boldsymbol{d}$ 的 n 重副本和 $\boldsymbol{q}$ 之间的相似性应和 $\boldsymbol{d}$ 本身和 $\boldsymbol{q}$ 之间的相似性相同：

$$\left\|\frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}-\frac{\boldsymbol{q}}{\|\boldsymbol{q}\|}\right\|$$

- 余弦相似性，即向量 $\boldsymbol{d}$ 和 $\boldsymbol{q}$ 之间角度的余弦：

$$\frac{\boldsymbol{d}\cdot\boldsymbol{q}}{\|\boldsymbol{d}\|\,\|\boldsymbol{q}\|}$$

也可以参见式 (15.3)。

因此，基于 TF-IDF 坐标的信息检索系统工作原理如下。首先建立 TF(t,d) 和 IDF(t) 信息的逆索引。给定一个查询时，将它映射到 TF-IDF 空间，根据相似性度量对文档进行排名，并返回最相似的文档。搜索方法可以不同的方式扩展，例如搜索短语。参考文献 [34] 中给出了这一主题的更多细节。

注意，传统的 TF-IDF 表示中没有什么神奇的：这只是一个启发式的方法，给信息量更丰富的单词更大的权重，使上面给出的标准度量能得出合理的结果。基于信息内容（如互信息）的更复杂的度量学习或特征选择方法会带来更好的结果，但需要用户提供更多的信息。

25.2.2 相关反馈

如上所述，将查询转化成向量后，可以用一个向量相似性度量来识别一组最相似的文档并返回给用户。

不幸的是，由于平常的网页查询只有一个词或两个词那么长，检索到很多不相关的内容也并不奇怪。因此，要么需要一个严肃度量来定性地排名文档（例如 25.3 节的 PageRank），要么至少以一种方式来快速地**从用户获得反馈**，并以此产生更好的查询。

罗基奥方法（Rocchio's Method）更新用于第一查询的向量，使之相似于被用户标识为相关（喜欢）的文档的描述向量，而区别于被列为不相干（不喜欢）的向量，如图 25-7 所示。这一过程可以想象为用户喜欢的文档吸引查询向量，不喜欢的文档排斥查询向量。具体来说，查询向量被更新为：

$$\boldsymbol{q}' = \alpha\boldsymbol{q} + \beta\sum_{\boldsymbol{d}\in D_+}\boldsymbol{d} - \gamma\sum_{\boldsymbol{d}\in D_-}\boldsymbol{d}$$

其中，D_+ 是一组用户喜欢的文档，D_- 是一组用户不喜欢的文档。参数 α、β 和 γ 控制改变的量。细心的读者可能会注意到，原型向量更新的方式与第 17 章中的自组织映射的方式相似。

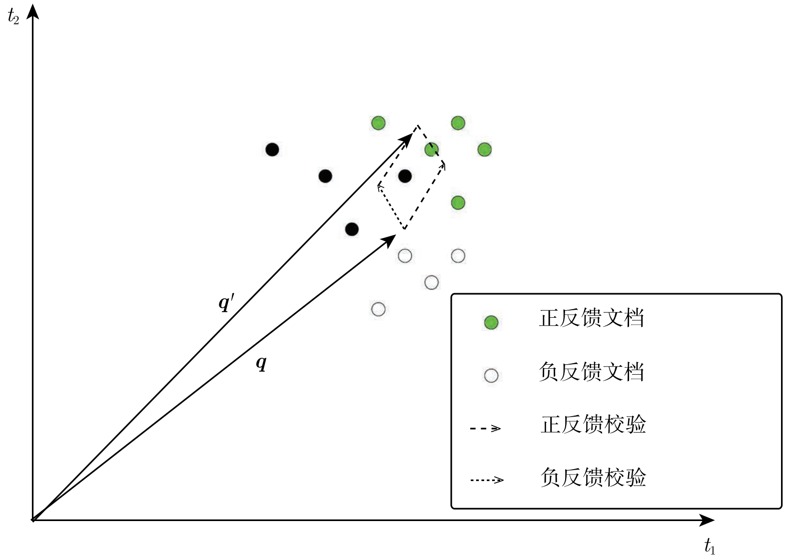


图 25-7　罗基奥方法

25.2.3　更复杂的相似性度量

在一个 TF-IDF 向量空间，可以用距离上的递减函数定义两个词之间的“相似性”为距离的逆。例如，尽管一个对象与自身进行比较时，相似性趋于无穷，但需要某些修正。如果这些元素用了集合表示法，可以用另一个相似性度量：杰卡德系数（Jaccard coefficient）。令 A 和 B 为两个（有限）集合，A 和 B 的杰卡德系数定义为

$$r'(A,B) = \frac{A\cap B}{A\cup B}$$

它的目标应该很清楚：将两个集合中的共同元素（交集）和它们的总规模进行比较。它在 0 和 1 之间取值，$r'(A,B)=0$ 表示两个集合中没有共同元素，$r'(A,B)=1$ 表示 A 和 B 是相等的。一个附加的重要性质是，$1-r'(A,B)$ 是一个距离，它服从一个度量的所有性质。

接下来采用一个更以文档为中心的定义。如果 d 是一个文档，那么定义 $T(d)$ 为它所包含的一组标记（词）。需要注意的是，由于集合中的元素一般来讲没有多重性（multiplicity），而且我们只对二进制模型感兴趣，即一个词或者存在，或者不存在。那么，这两个文档的杰卡德系数是

$$r'(d_1, d_2) = \frac{|T(d_1) \cap T(d_2)|}{|T(d_1) \cup T(d_2)|}$$

在搜索引擎中使用的的杰卡德系数基于以下事实：用户所认为的查询通常是一组不重复的词，没有用户会在谷歌里输入类似“reactive reactive search”的查询，然后期盼返回包含单词“reactive”两倍于单词“search”的文档。

下面是一种计算杰卡德系数 $r'(\cdot,\cdot)$ 的算法框架。

- 对每一个 $d \in D$
  - 对每一个词 $t \in T(d)$：将记录 (t, d) 存进文件 f_1。
- 以 (t, d) 的顺序对 f_1 进行排序，然后将其合并成 (t, D_t) 的形式。
- 对每一个在 f_1 扫描到的词 t
  - 对每一对 $d_1, d_2 \in D_t$：将记录 (d_1, d_2, t) 存进文件 f_2。
- 以 (d_1, d_2) 的顺序对 f_2 进行排序，然后将其加入第三个字段进行合并。

一些技巧可降低搜索成本，比如可以预先为所有文档查询对计算杰卡德系数，否则需要大量的存储和 CPU 时间；或者通过给每个文档或者查询预关联一个小而数量固定的最相似文档的列表，减少文档查询对的数量。此外，非常频繁的词（低 IDF）可以完全不考虑。

在实践中，许多情况下人们对近似（approximating）系数感兴趣。一个可行的方式是利用随机排列的有趣的随机算法。

如果使用概率，给定集合 A 和 B，我们从这个有趣的等式开始：

$$\frac{|A \cap B|}{|A \cup B|} = \Pr(x \in A \cap B | x \in A \cup B)$$

如果可以估计上述概率，就可以估算杰卡德系数。我们所能做的是从集合 $S \subset \{1, \cdots, n\}$ 生成随机元素，并通过事件的比值来估算概率。

从集合 $S \subset \{1, \cdots, n\}$ 中选择随机元素，可以选择 n 个元素的随机排列 π，并在 S 中挑选元素，使得它在 π 中的像是最小的：

$$x = \arg\min_{x \in S} \pi(x) = \arg\min \pi(S)$$

当应用于 $A \cup B$ 时，该方法在交集中定位一个元素，当且仅当：

$$\min \pi(A) = \min \pi(B)$$

因此，可以通过随机排列并检查两个最小值是否重合来估计上述比例。

接下来证明为什么排列是有效的。给定集合 $A, B \subset \{1, \cdots, n\}$，为了得出两个最小值重合的概率，建立一个排列，计算有多少排列 $\pi : \{1, \cdots, n\} \to \{1, \cdots, n\}$（$n!$ 个可能性）有下述性质：

$$\min \pi(A) = \min \pi(B)$$

从图 25-8 中应当搞清楚的是：

- $A \cup B$（灰块和斜纹块）的像有 $\binom{n}{|A \cup B|}$ 种不同的选择方式；
- 在这样的像中，可以在组成交集的 $|A \cap B|$ 个元素中选取最小元素（粗箭头）；
- $A \cup B$ 的像中的其余元素可以有 $(|A \cup B| - 1)!$ 种排列方法（细箭头）；
- 不在 $A \cup B$ 内的元素有 $(n - |A \cup B|)!$ 种排列方法（虚线箭头）。

所有这些相乘之后，可以得到：

$$\binom{n}{|A \cup B|} \cdot |A \cap B| \cdot (|A \cup B| - 1)! \cdot (n - |A \cup B|)! = n! \frac{|A \cap B|}{|A \cup B|}$$

除以排列的总数后，推导出所需的等式。

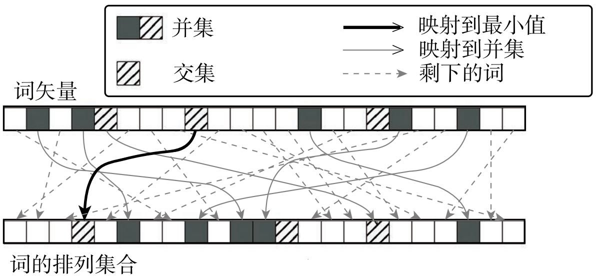


图 25-8　构建一个排列

一个随机但效率低下的算法如下：

- 生成词集上的 m 个排列的集合 Π；
- $k \leftarrow 0$；
- 对每一个 $\pi \in \Pi$
  - 若 $\min \pi(T(d_1)) = \min \pi(T(d_2))$，则 $k \leftarrow k + 1$；
- 估计 $r'(d_1, d_2) \approx \dfrac{k}{m}$。

通过将随机算法与多文档系数的同步计算相结合，就能够处理万维网中的大量文档，其中随机算法使用适用于外部存储的数据结构。

25.3　使用超链接来进行网页排名

如前所述，网页是如此之多，以至于检索一些与查询相关的页面的集合已远远不够，还

要检索高质量的相关网页集合。这个问题早在网页出现之前就有了，阅读书籍或论文时也会遇到。人们想把精力花在高质量的某一主题的论文上，不会在质量差的论文上浪费时间。在科学界，如果一篇论文被其他高质量的论文引用，那么这篇论文也会被认为是高质量的，因为这意味着有些同行发现这篇论文有用，并通过把该论文加入到引文列表中以表示认可。一个更现实的比喻是，某个应聘者是有价值的，如果有许多其他有价值的人准备推荐他。一般情况下参见图 25-9，在人与人之间的社交网络中，一个人的声誉是通过其他拥有高度声誉的人推荐而获得的。说服许多声誉不够高的人支持你是不够的，没有捷径可走！

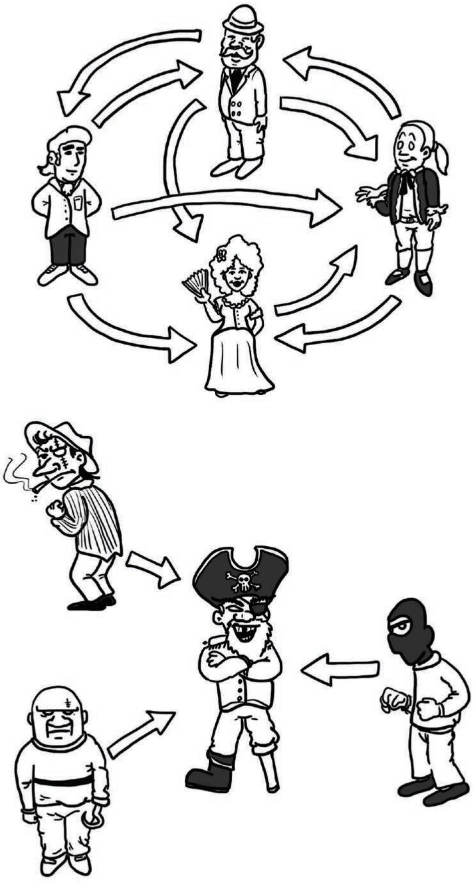


图 25-9　社交网络中的声望：被声望高的人推荐（或者有关联，上图）比被声望低的人推荐（下图）更容易使一个人获得高声望

马尔基奥里一篇开创性的论文 [81] 强调了超信息（超链接中的信息）的重要性后，拉里佩奇和谢尔盖　布林开发了 PageRank 算法，这一算法遵循同样的基本社交网络原则，用超链接代替了“建议”和“引用”[84]（作者后来成为了谷歌创始人）。他们定义了“声望度量”，

使得网页的声望与链接到它的声望高的网页的数量相关。值得注意的是，这是一个递归定义。为了衡量一个页面的声望，需要知道指向它的网页的声望等信息。简而言之，他们的解决方案是：先从一个声望值的初始分布开始，迭代计算不同节点的声望值，直到重新计算后的声望值变化很小为止，就这么简单！这种方法乍一看会让人觉得不太可靠。是什么能够保证，无论采用什么初始分布，总能收敛到相同的最终分布？

现在令人着迷的是，基本线性代数的**本征值和本征向量**的概念，以及**马尔可夫链**的相关概念是怎样与这一问题的解决方案关联起来的。下面来总结一下主要关系。

首先来看看，从初始分布入手迭代地计算声望，与**计算矩阵的主本征向量的经典幂迭代法**是如何相关的。这里讲得很快，跳过数学细节，只为了让大家对该方法有一些体会。

检查链入链接（从其他页面指向给定页面的超链接），计算出网页的排名。每个网页 i 的链入链接为排名做出部分贡献，等于 i 的排名除以其链出链接的数目，如图 25-10 所示。除法的动机很明显：一个排名很高的网页指向数量庞大的页面，就像一个人很有声望，但推荐的人太多。如果没有除法，那么排名居首位的网页的所有者就可以仅靠投放大量的链出链接影响全世界的所有页面。

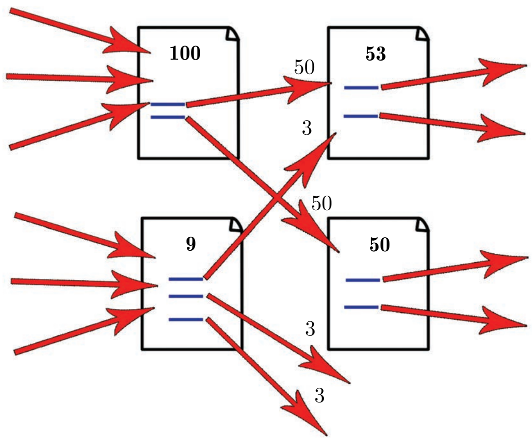


图 25-10　在 PageRank 算法中重新计算网页的排名，初始排名沿着链出链接分发到其他网页（改编自原始论文）

鉴于上述重新计算的规则，一旦给出了超链接的网络，第 k 次迭代时的新排名值 $\boldsymbol{p}^k$，就能通过矩阵 $\boldsymbol{M}$ 对前一次的值进行线性变换而计算得到，如下所示：

$$\boldsymbol{p}^k = \boldsymbol{M}\boldsymbol{p}^{k-1}$$

矩阵 $\boldsymbol{M}$ 仅依赖于链接结构，即页面之间的链接。现在，从初始排名分布 $\boldsymbol{p}^0$ 开始，执行 k 次重新计算：

$$\boldsymbol{p}^k = \boldsymbol{M}^k\boldsymbol{p}^0$$

假定存在 $\boldsymbol{M}$ 的本征向量的一组基向量，令 $\lambda_1, \lambda_2, \cdots, \lambda_n$ 为 n 个本征值，并令 $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n$ 为相应的本征向量。假设 λ_1 是主本征值，那么对于所有 $j > 1$，有 $|\lambda_1| > |\lambda_j|$。

初始向量 $\boldsymbol{p}^0$ 可以写成一组基向量的线性组合：

$$\boldsymbol{p}^0 = c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \cdots + c_n\boldsymbol{v}_n$$

如果 $\boldsymbol{p}^0$ 是随机选择的（以均匀概率抽取），那么 $c_1 \neq 0$ 的概率为 1。现在，使用线性性和本征向量的定义性质，可以马上得到：

$$\begin{aligned}\boldsymbol{M}^k\boldsymbol{p}^0 &= c_1\boldsymbol{M}^k\boldsymbol{v}_1 + c_2\boldsymbol{M}^k\boldsymbol{v}_2 + \cdots + c_n\boldsymbol{M}^k\boldsymbol{v}_n \\ &= c_1\lambda_1^k\boldsymbol{v}_1 + c_2\lambda_2^k\boldsymbol{v}_2 + \cdots + c_n\lambda_n^k\boldsymbol{v}_n \\ &= c_1\lambda_1^k\left(\boldsymbol{v}_1 + \frac{c_2}{c_1}\left(\frac{\lambda_2}{\lambda_1}\right)^k\boldsymbol{v}_2 + \cdots + \frac{c_n}{c_1}\left(\frac{\lambda_n}{\lambda_1}\right)^k\boldsymbol{v}_n\right)\end{aligned}$$

当重新计算的次数 k 与矩阵 $\boldsymbol{M}^k$ 的幂值相等，趋于无穷大时，除了一个正比于主本征向量的项，其他所有项都趋于零。几乎所有从任意初始条件出发的矩阵乘法的简单迭代法，都确实足以提取主本征向量!

现在考虑一个不同的解释，与马尔可夫链有关。想象一下，对一个系统进行分析，这一系统表示用户在不同网页上**浏览的移动**。假设用户永远通过链出链接导航，并随机均匀地选择。令起始页 u 浏览的行为 p_u^0。令 $\boldsymbol{E}$ 为网页的邻接矩阵：$(u, v) \in \boldsymbol{E}$（或 $\boldsymbol{E}_{uv} = 1$），当且仅当存在从页面 u 到页面 v 的链接。浏览者点击 i 次后到达页面 v 的概率 p_v^i 是什么？

下面从单一的步骤开始。浏览者点击 1 次后到达页 v 的概率 p_v^1 是什么？令

$$N_u = \sum_v \boldsymbol{E}_{uv}$$

为页面 u 的**出度**（$\boldsymbol{E}$ 中第 u 行的和）。假设不存在平行边，

$$p_v^1 = \sum_{(u,v)\in\boldsymbol{E}} \frac{p_u^0}{N_u}$$

通过归一化 $\boldsymbol{E}$ 来使得行之和为 1：

$$L_{uv} = \frac{\boldsymbol{E}_{uv}}{N_u}$$

得到

$$p_v^1 = \sum_u L_{uv}p_u^0 \qquad \text{或} \qquad \boldsymbol{p}^1 = L^{\mathrm{T}}\boldsymbol{p}^0$$

现在来考虑第 i 步之后的情况：

$$\boldsymbol{p}^i = L^{\mathrm{T}}\boldsymbol{p}^{i-1}$$

如果 $\boldsymbol{E}$ 是不可约的（irreducible）且非周期的（aperiodic）（没有一个是真正成立的，但这个问题可以解决），那么：

$$\lim_{i\to\infty} \boldsymbol{p}^i = \boldsymbol{p}$$

其中 $\boldsymbol{p}$ 是 L^{T} 的主本征向量，也称作它的*稳定分布*（stationary distribution）：

$$\boldsymbol{p} = L^{\mathrm{T}}\boldsymbol{p} \qquad （本征值为 1）$$

但 p_u 是页面 u 的*声望*（prestige），也由前面的解释所确定。现在应该很清楚，声望也可以解释为在一个给定页面沿着链接找到随机浏览者的概率。

现在来处理真实世界过渡矩阵的不好的性质。有研究显示，网页之间并没有强大的连接，并且其中的随机游走可能陷入循环。一个可能的解决方法是引入“阻尼系数”，它反映了用户可能偶尔停止跟随链接：在每一步以任意概率 d 访问一个随机页面（甚至是不相连的）。因此，过渡可以表示为：

$$\boldsymbol{p}^i = \left((1-d)L^{\mathrm{T}} + \frac{d}{N}\mathbf{1}_N\right)\boldsymbol{p}^{i-1}$$

矩阵最大本征值对应的本征向量可以用如下方式得到。

- 从一个随机向量 $\boldsymbol{p} \leftarrow \boldsymbol{p}^0$ 开始
- 重复
  - 更新向量：
  $$\boldsymbol{p} \leftarrow \left((1-d)L^{\mathrm{T}} + \frac{d}{N}\mathbf{1}_N\right)\boldsymbol{p}$$
  - 不时地进行归一化：
  $$\boldsymbol{p} \leftarrow \frac{\boldsymbol{p}}{\|\boldsymbol{p}\|_1}$$

归一化是为了避免出现十分大的分量，因此不会出现有限精度计算的数值问题。当然，在这一应用中，我们感兴趣的不是绝对声望值，而是相对声望值。其绝对值取决于所选取值的范围（可以选择 0~10 来衡量声望，也可以选择 0~100，等等），而相对值是指，比如，某网页的声望比另一网页的声望高出 3 倍。归一化是减少特征向量倍数影响的一种简单方法。

在实际的应用中，声望的概念是如此模糊，以至于没人期望能以高精度得到实际的本征向量！为了尝试需要多久才能收敛，佩奇在其原始论文上说，3×10^8 个页面的情况下，52 次迭代已经足够了，这是一个相当令人振奋的结果，为重要的商业应用铺平了道路。

25.4　确定中心和权威：HITS

现在考虑一种不同的网页分析。在科学界，好的文章要么影响深远（即被许多其他文献参考），要么属于综述（即引用许多其他文献）。同理，网页也可分为*权威*（authority）或*中心*

（hub）两类 [47]。例如门户网站是很好的中心，尽管它们不包含重要的信息，只是达到高质量网页的起始点。

为了反映这一区别，接下来介绍两种度量，称为**中心度**（表示为 $\boldsymbol{h}$）和**权威性**（表示为 $\boldsymbol{a}$）：

$$\boldsymbol{h} = (h_u), \qquad \boldsymbol{a} = (a_u)$$

现在来总结一下 HITS 算法（超链接诱导主题搜索）。在 HITS 算法中，第一步是检索搜索查询的结果集。给定查询 q，令 R_q 为一个 IR 系统所返回的根集。计算仅在此结果集上进行，而不是所有网页。权威性和中心度相互递归定义。

通过添加链接到根的所有节点组成扩大集：

$$V_q = R_q \cup \{u : ((u \to v) \vee (v \to u)) \wedge v \in R_q\}$$

令 E_q 为诱导链接子集，$G_q = (V_q, E_q)$。递推关系定义如下。令权威值 h_u 与被引用的权威页面总数成正比，令中心度 a_u 与被引用的中心页面总数成正比。

$$\boldsymbol{a} = E^{\mathrm{T}}\boldsymbol{h}$$

$$\boldsymbol{h} = E\boldsymbol{a}$$

因此给出下列迭代方法：

- 初始化 $\boldsymbol{a}$ 和 $\boldsymbol{h}$（例如，以均匀概率）；
- 重复
  - $\boldsymbol{h} \leftarrow E\boldsymbol{a}$；
  - $\boldsymbol{a} \leftarrow E^{\mathrm{T}}\boldsymbol{h}$；
  - 归一化 $\boldsymbol{h}$ 和 $\boldsymbol{a}$。

排名靠前的权威页面和中心页面报告给用户。

主本征向量识别最大的密集二部子图。若要找到较小的集合，则必须探索其他本征向量。存在从一个系统中删除已知本征向量的迭代方法：当确定了一个本征向量时，就减少搜索子空间。

虽然对于理论是有价值的，实际中搜索引擎不常使用 HITS，再加上实践中为不同的查询预先计算中心度和权威值是不可行的，该算法必须在查询执行之后运行，使得该算法对于通用目的而言十分笨重。回到 PageRank 算法，注意它是独立于页面内容的，因此必须有机地结合内容，具体取决于执行的查询。谷歌组合查询和排名的方式是未知的。也许，经验参数和人工检查是必要的。

25.5　聚类

聚类的原因是网页搜索中检索文档的数量巨大。为了避免用户过载，识别密切相关的文档组是有用的，例如仅显示少量具有代表性的原型。

自动识别的聚类也可以为后续的人工分类提供帮助。此外，如果用户对文档 d 感兴趣，那么他很可能也对相同聚类里的文档感兴趣。因此，预先计算聚类，使得用户在因需求检索时获得更多相似的文档。

注意，查询可以是模糊的，尤其是网页查询。例如，如果搜索“星”，人们可能在寻找电影明星，也可能是天体，这显然是两个区别很大的主题。

词向量空间里的相互相似性有助于将相似文档分在一起，也就是说，找到文件“聚类”。设 D 为根据相似性组合在一起的文档（或其他实体）集。条目 $d \in D$ 的特征要么是一些内部的固有性质（例如包含的词，以及 TF-IDF 空间中的坐标），要么是通过外部的距离度量 $\delta(d_1, d_2)$ 或对 $\rho(d_1, d_2)$ 之间的相似性。实例有欧氏距离、点积、杰卡德系数。定义了度量之后，可以使用通常的自底向上或自顶向下的聚类技术。这些方法已经在第 15 章和第 16 章介绍过。



梗　概

网页包含类型广泛的数据，有些是有结构的，有些是有部分结构的，还有些是完全没有结构的。**爬虫和索引**都是系统的方法，用以访问网页、收获其中所载的信息，以及为搜索、信息检索和排名准备数据结构。

通过将文本转换成数据向量（例如**向量–空间模型**中的所选单词的频率），可以重用传统的 ML 技术，但是网页文档中丰富的结构适用于一些更专门的分析。

网页挖掘技术查找文档（网页链接）之间的显式关系、推断隐式关系（通过聚类）、在连接站点的网络上排名最相关的页面，或在社交网络中识别最相关和人脉最广的人。抽象使我们能对网页和社交网络使用类似的工具。一个值得注意的例子是，超链接和线性代数工具（本征向量和本征值）的使用，过去曾在文献计量学中给研究者排名，如今带来了非常强大的**网页排名**技术，也是谷歌的搜索引擎技术的基础。

从今往后，你再看到超链接、Facebook 上的“赞”和 Twitter 上的“粉丝”（或者你读到这本书时最流行的社交软件）时，你会用不同以往的分析思维和观点来看待它们。

第 26 章　协同过滤和推荐

诽谤是微风，温柔的和风，
它不知不觉地、巧妙地、轻轻地和甜甜地，
开始窃窃私语。

—— 罗西尼，《塞维利亚的理发师》



口口相传一直是一种强大而有效的技术，在人与人之间病毒式传播信息和观点。它以分布式和人力的方式来**挖掘隐藏在人脑海中的数据**。它是有效的，因为我们自然倾向于与相似的人说话，分享生活的习惯、观点和方式。通过与挑选出来的少数人进行交流，我们可以有效地**过滤**数据。然后，由我们来决定、整合和权衡接收到的信息，并做出最终决定。类似的过程可以通过数据挖掘和建模方法来模拟。人们从原始数据中（数量巨大，范围从几千到十亿项）提取信息，它们与特定的最终用户相关，基于其显式或隐式的偏好模型，以及与其他人的相似性。

一个有趣的应用是在营销部门：收集用户和产品的相关数据，无论是购买过的还是仅评价过的，这些数据可以用于估计客户将如何评估一个之前没有见过的产品。预测评估的最终目的是鼓励用户购买，例如推荐相应的预测评价最高的物品清单。如果是基于用户的偏好过滤所呈现的产品，那么广告会更奏效。另一个应用是网页挖掘，目标是在搜索信息时确定用

户可能感兴趣的网页。因此，目的在于**让信息仿照口口相传**的模式扩散，无论是正面（赞誉）还是反面（诽谤）意见。

协同过滤和推荐是通过从许多其他合作者收集品味信息来预测一个人的兴趣点的方法。一个潜在的假设是，**过去兴趣一致的人，将来的兴趣也倾向于一致**。例如，给定关于用户品味的一些信息以及从许多其他用户那里收集的信息，电影的协同推荐系统可**预测用户**个人喜欢哪些电影。

表 26-1　评分矩阵

	电影 1	电影 2	电影 3	电影 4
用户 1	1	4	2	1
用户 2	1	5	1	0
用户 3	1	0	0	0

考虑一个用户–项目矩阵 $\boldsymbol{R}$，其中每一个元（entry）r_{ui} 的值是用户 u 对项目 i 的评分，如表 26-1 所示。每个用户 u 可以选出物品 i 在区间 [最低分数, 最高分数] 中的分数。具体来说，假设最低分数 $= 1$，最高分数 $= 5$，值 0 表示未知分数。想预测矩阵中未知的分数，要么通过一些直接的方式，要么通过更紧凑的方式来表示数据，并通过该紧凑表示来预测。

26.1　通过相似用户结合评分

有一个简单的方法来预测未知的评分 r_{ui}，考虑其他用户对同一个项目 i 的评分，以及用户 u 和其他用户的相似性。一般的未知评分 r_{ui} 由如下公式计算：

$$r_{ui} = \frac{\displaystyle\sum_{\text{已知}r_{ki}} \text{相似性}(u,k) \cdot r_{ki}}{\displaystyle\sum_{\text{未知}r_{ki}} \text{相似性}(u,k)} \tag{26.1}$$

该用户的评分通过其他用户评分的加权平均值来预测，权重由相似性给出，如图 26-1 所示。其原因是相似的用户往往给出相似的评分。若上述式子的分母为 0，则 r_{ui} 默认计算为所有已知评分 r_{ki} 的平均值。若项目 i 没有人评分，则 r_{ui} 为 0。

以类似的方式，可以对同一用户对不同项目的评分求平均值，权重与项目之间的相似性成正比，如图 26-1 下半部分所示（相似的项目往往以相似的方式来判断）。

问题的关键是如何度量相似性。在此简化的背景下，关于用户的仅有的知识必须通过他对过去不同项目的评价得到。因此，在式 (26.1) 中，两个用户 (u,k) 之间的相似性通过度量两个向量 $(\boldsymbol{v}_u, \boldsymbol{v}_k)$ —— 评分矩阵 $\boldsymbol{R}$ 的第 u 行和第 k 行 —— 之间的相似性得到。

在标准实现中，可以使用两个向量之间通常的余弦相似性，但如 15.2 节所说的，也可以检验不同的针对具体问题的度量。

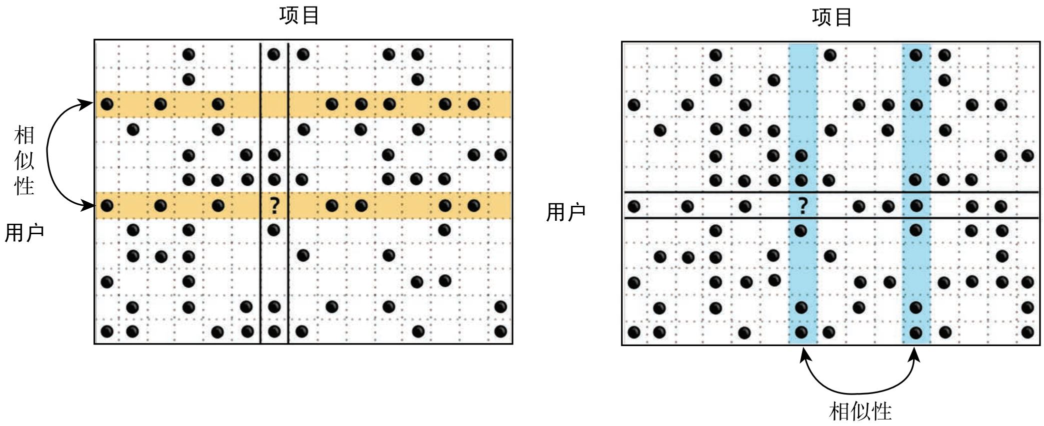


图 26-1 协同推荐。要预测未知值，可以计算已知值的加权平均，权重取决于用户或项目之间的相似性

说实话，人们表达意见的方式非常不同。对于同一部电影，一个保守的英国评论者的评价是“尚可”，而一个夸张的意大利评论者的评价则可能是“梦幻般的电影”。如果你听从某个十分挑剔的评论者的观点，最终你就没什么电影可看了，因此在用这些评价度量相似性并进行预测之前，**减少个体评估的作用**可能是有用的。

令 r_{uj} 是用户 u 对物品 j 的评分。令 I_u 是用户 u 已评分的项目的集合。用户 u 的平均评分为 $\bar{r}_u = \frac{1}{|I_u|}\sum_{j \in I_u} r_{uj}$。活跃用户用下标 a 表示。目标是预测用户对项目 i 的偏好，或 p_{ai}。

因为评分可能不以零为中心，所以该系统较难通过标量积再现这些评分。为了帮助该系统，也可以减去平均值，从而得到中心化的数据。具体来说，预测可以通过下式计算：

$$p_{ai} = \bar{r}_a + \frac{\sum_u w_{au}(r_{ui} - \bar{r}_u)}{\sum_u |w_{au}|} \tag{26.2}$$

其中对 u 求和是指评价过项目 i 的用户集合，而 w_{au} 是活跃用户 a 和用户 u 之间的权重。该权重可定义为皮尔逊相关系数：

$$w_{au} = \frac{\sum_i (r_{ai} - \bar{r}_a)(r_{ui} - \bar{r}_u)}{\sqrt{\sum_i (r_{ai} - \bar{r}_a)^2}\sqrt{\sum_i (r_{ui} - \bar{r}_u)^2}} \tag{26.3}$$

对 i 求和是指集合 $I_u \cap I_a$。

26.2　基于矩阵分解的模型

原始的用户–项目评分矩阵的**稀疏性**（sparsity）可能是个问题。每个用户评价的项目只是一个很小的子集，大多数项目的评分是未知的。通过将**用户特征压缩和总结**到小得多的向量，有望得到更好的泛化结果，并能更好地理解模型，正如奥卡姆剃刀原理所解释。

有一种确定用户对于某个项目的兴趣或者评分的可行方法，是**给每个用户和每个项目都关联一个特征向量**（vector of characteristics），然后通过观察用户和项目的特征向量之间的相似性推测出评分。该操作可以人工来完成，但可能非常耗时，也可能无法识别对于预测至关重要的某些特征。接下来看看这一过程是如何实现自动化。

用向量 $\boldsymbol{q}_i \in \Re^f$ 表示项目 i 的特征（因子），向量 $\boldsymbol{p}_u \in \Re^f$ 表示用户 u 对此项目的每个因数感兴趣的程度。通过简单计算相应向量的标量积，就能得到用户 u 对项目 i 的评分：

$$\hat{r} = \boldsymbol{q}_i^{\mathrm{T}} \boldsymbol{p}_u \tag{26.4}$$

例如，如果人工建立这些因子，电影《终结者》的权重可以是（动作 = 5, 浪漫 = 1），用户帕特里夏对电影的兴趣是（动作 = 2, 浪漫 = 5），因此用户帕特里夏对电影《终结者》的评分是 $5 \cdot 2 + 1 \cdot 5 = 15$。

在通过建立有效因子来预测评价的自动化方式中，传统的奇异值分解（SVD）可用于查找有效的 $\boldsymbol{q}_i$ 和 $\boldsymbol{p}_u$。使用 SVD 可将包含所有评分的矩阵 $\boldsymbol{R}$ 分解为 $\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{M}^{\mathrm{T}}$，其中矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{M}$ 的行是 $\boldsymbol{p}_u$ 和 $\boldsymbol{q}_i$ 的集合，由对角矩阵 $\boldsymbol{\Sigma}$ 进行缩放。通过调整 $\boldsymbol{\Sigma}$ 的对角线值的大小，可以降低向量的维数，并且仅保留最相关的部分。不幸的是，大多数情况下，无法得到评分矩阵所有单元的值。

基于优化的更加灵活和健壮的学习算法，可以用来找到因子向量 $\boldsymbol{q}_i$ 和 $\boldsymbol{p}_u$ 有效的近似值。像往常一样，表达评分的实例引导学习过程。为了学习因子向量 $\boldsymbol{q}_i$ 和 $\boldsymbol{p}_u$，需要最小化已知评分集合上的正则化平方误差（RSE）：

$$\mathrm{RSE} = \frac{1}{|K|} \sum_{(u,i)\in K} (r_{ui} - \boldsymbol{q}_i^{\mathrm{T}} \boldsymbol{p}_u)^2 + \lambda(\|\boldsymbol{q}_i\|^2 + \|\boldsymbol{p}_u\|^2) \tag{26.5}$$

其中，K 是已知 r_{ui}（训练集）的 (u,i) 对的集合。要知道，求和的第一项是模型 $\boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{p}_u$ 与已知结果 r_{ui} 之间的平方误差。想要促进泛化（预测新评价），通过正比于常数 λ 的方式来惩罚因子向量的大小是有用的。这一项叫作**正则项**。当评分实例十分丰富时，大部分 RSE 的贡献来源于重建表达评分的误差。另一方面，当评分稀少时，正则项就变得至关重要，它的作用是避免出现非常大的向量，这些非常大的向量会给预测带来潜在的巨大（和错误的）影响。

现在的问题是最小化自由参数 $\boldsymbol{p}_u$ 和 $\boldsymbol{q}_i$ 的一个连续函数，第 21 章中说明的方法可以用在这里，例如传统的梯度下降法。RSE 的梯度计算如下：

$$\frac{\partial \mathrm{RSE}}{\partial \boldsymbol{q}_i} = \frac{2}{|K|} \sum_{(u,i)\in K} ((r_{ui} - \boldsymbol{q}_i^{\mathrm{T}} \boldsymbol{p}_u)(-\boldsymbol{p}_u) + \lambda \boldsymbol{q}_i)$$

$$\frac{\partial \text{RSE}}{\partial \boldsymbol{p}_u} = \frac{2}{|K|}\sum_{(u,i)\in K}((r_{ui} - \boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{p}_u)(-\boldsymbol{q}_i) + \lambda \boldsymbol{p}_u)$$

项 $1/|K|$ 是一个常数，不会影响最小化的结果。可以从 $\boldsymbol{q}_i$ 和 $\boldsymbol{p}_u$ 的随机初始值开始，然后进行迭代：在每一步中，沿着负梯度方向稍微进行改变，以减小 RSE 误差。

更精确的模型：加入偏差

如 26.1 节所示的简单的方法的情况，用户 u 对项目 i 的评分并不仅依赖于两个向量 $\boldsymbol{p}_u$ 和 $\boldsymbol{q}_i$ 之间的互动 $\boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{p}_u$，还依赖于用户或项目的偏差。换句话说，有的人往往给予较高的评价，而有的项目常常得到比别的项目更高的评价。关于评价 r_{ui} 的偏差可以描述为 $b_{ui} = \mu + b_i + b_u$，其中 μ 是总体平均值，b_i 和 b_u 是用户 u 和项目 i 分别观察到的平均偏差。例如，假设想要估计用户乔对电影《泰坦尼克号》的评价。假设所有电影的平均得分 μ 为 3.7 星。此外，《泰坦尼克号》比普通电影更好，因此它比平均值高出 0.5 星。另一方面，乔是一个苛刻的用户，常常给出低于平均分 0.3 星的分值。因此，乔对《泰坦尼克号》的评分的基准估计是 3.9 星（$3.7 + 0.5 - 0.3$）。

根据这个改进的模型，估计用户 u 对项目 i 的评分 $\hat{r}_{ui}$ 的计算公式为：

$$\hat{r}_{ui} = \mu + b_i + b_u + \boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{p}_u \tag{26.6}$$

观察到的评分被分解为 4 个组成部分：*全局平均值、项目偏差、用户偏差，以及用户–项目交互*。这使得每个组成部分仅解释评价中与其相关的一部分。学习算法通过最小化下面考虑偏差因子的正则化平方误差函数（RSEB）进行学习：

$$\text{RSEB} = \frac{1}{|K|}\sum_{u,i\in K}(r_{ui} - \mu - b_i - b_u - \boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{p}_u)^2 + \lambda(\|\boldsymbol{p}_u\|^2 + \|\boldsymbol{q}_i\|^2 + b_i^2 + b_u^2) \tag{26.7}$$

通常默认程序从推导梯度和使用最速下降开始。使用最速下降的一个分解过程见图 26-2：随着梯度下降的迭代步数增加，正如所料，训练集上的误差（均方根误差，RMSE）减少。在测试集（训练期间没有使用的评分）上，误差先减少，但随后达到一个平台，最终逐渐增加。这是**过拟合**的一个实例：系统试图精确地再现训练实例，但泛化能力却变差了。想想一个学生通过死记硬背学习，如果不理解学习材料，也无法提取相关关系。

注意**优化的力量和灵活性**：如果将附加项添加到模型里，可立即通过计算新的偏导数并将其插入最小化算法来确定最佳参数。如果知道如何优化，就可以专注于问题的定义，然后迅速尝试很多可供选择的模型，以及在验证数据上测试得到的泛化结果。

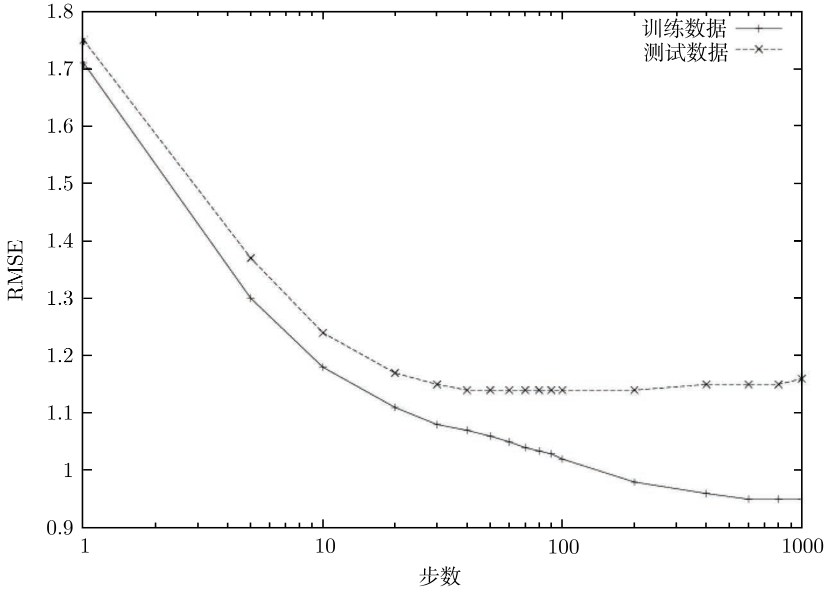


图 26-2　一个实际的分解过程：训练和测试性能表示为梯度下降的迭代步数的函数



梗　　概

当潜在客户访问您的电商门户网站时，他会查看一些商品，加入购物车并购买，撰写评论并打分，留下了痕迹和气味，训练有素的“鼻子”能够探测到。

所有这些信息都能帮助你提升你的服务：就像一个好的店主，他能叫出顾客的名字，并主动展示他们可能最喜欢的东西，个性化定制的商品展示能让你的网站更吸引客户。

协同过滤正是这方面的选择：通过记忆和分析客户的行为，能够同步描绘访问者和商品特征，通过类似的购物习惯对人们进行分组，并**预测客户可能最喜欢哪些商品**。这种个性化的过程不需要特定的领域知识，只需要挖掘客户的集体行为。这就是一个书呆子教授最终可以胜任复杂的时尚业务顾问的原因。

现在，在点击你喜爱的在线报纸的一个八卦标题之前请三思。如果你这样做，那么将有越来越多的八卦新闻出现在你的个性化主页里（也许还会出现在你访问的不同网站里，这是分享营销数据和行为重定向策略所导致的）。

参 考 文 献

[1] Y.S. Abu-Mostafa. Learning from hints in neural networks. *Journal of Complexity*, 6(2): 192–198, 1990.

[2] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive science*, 9(1): 147–169, 1985.

[3] Horace B Barlow. Summation and inhibition in the frog's retina. *The Journal of physiology*, 119(1): 69–88, 1953.

[4] R. Battiti. Accelerated back-propagation learning: Two optimization methods. *Complex Systems*, 3(4): 331–342, 1989.

[5] R. Battiti. First-and second-order methods for learning: Between steepest descent and newton's method. *Neural Computation*, 4: 141–166, 1992.

[6] R. Battiti. Using the mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4): 537–550, 1994.

[7] R. Battiti and M. Brunato. Reactive search: machine learning for memory-based heuristics. In Teofilo F. Gonzalez, editor, *Approximation Algorithms and Metaheuristics*, chapter 21, pages 21–1–21–17. Taylor and Francis Books (CRC Press), Washington, DC, 2007.

[8] R. Battiti and M. Brunato. Reactive Search Optimization: Learning while Optimizing. *Handbook of Metaheuristics*, 146: 543–571, 2010.

[9] R. Battiti, M. Brunato, and F. Mascia. *Reactive Search and Intelligent Optimization*, volume 45 of *Operations research/Computer Science Interfaces*. Springer Verlag, 2008.

[10] R. Battiti and P. Campigotto. Reactive search optimization: Learning while optimizing. an experiment in interactive multi-objective optimization. In S. Voss and M. Caserta, editors, *Proceedings of MIC 2009, VIII Metaheuristic International Conference*, Lecture Notes in Computer Science. Springer Verlag, 2010.

[11] R. Battiti and A. M. Colla. Democracy in neural nets: Voting schemes for accuracy. *Neural Networks*, 7(4): 691–707, 1994.

[12] R. Battiti and F. Masulli. BFGS optimization for faster and automated supervised learning. In *Proceedings of the International Neural Network Conference (INNC 90)*, pages 757–760, 1990.

[13] R. Battiti and A. Passerini. Brain-Computer Evolutionary Multiobjective Optimization (BC-EMO): A Genetic Algorithm Adapting to the Decision Maker. *IEEE Transactions on Evolutionary Computation*, 14(15): 671–687, 2010.

[14] R. Battiti and G. Tecchiolli. Learning with first, second, and no derivatives: a case study in high energy physics. *Neurocomputing*, 6: 181–206, 1994.

[15] R. Battiti and G. Tecchiolli. The reactive tabu search. *ORSA Journal on Computing*, 6(2): 126–140, 1994.

[16] R. Battiti and G. Tecchiolli. Training neural nets with the reactive tabu search. *IEEE Transactions on Neural Networks*, 6(5): 1185–1200, 1995.

[17] R. Battiti and G. Tecchiolli. The continuous reactive tabu search: blending combinatorial optimization and stochastic search for global optimization. *Annals of Operations Research-Metaheuristics in Combinatorial Optimization*, 63: 153–188, 1996.

[18] Roberto Battiti and Alan Albert Bertossi. Greedy, prohibition, and reactive heuristics for graph partitioning. *IEEE Transactions on Computers*, 48(4): 361–385, Apr 1999.

[19] Roberto Battiti and Anna Maria Colla. Democracy in neural nets: Voting schemes for classification. *Neural Networks*, 7(4): 691–707, 1994.

[20] A.B. Baum. On the capabilities of multilayer perceptrons. *Journal of Complexity*, 4: 193–215, 1988.

[21] Yoshua Bengio. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2(1): 1–127, 2009.

[22] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

[23] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2): 157–166, 1994.

[24] M. Bilenko, S. Basu, and R.J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 11. ACM, 2004.

[25] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3): 268–308, 2003.

[26] C.G.E. Boender and A.H.G. Rinnooy Kan. A bayesian analysis of the number of cells of a multinomial distribution. *The Statistician*, 32: 240–249, 1983.

[27] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[28] Leo Breiman. Random forests. *Machine learning*, 45(1): 5–32, 2001.

[29] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and regression trees*. Chapman & Hall/CRC press, 1993.

[30] C. G. Broyden, J. E. Dennis, and J. J. More. On the local and superlinear convergence of quasi-newton methods. *J.I.M.A*, 12: 223–246, 1973.

[31] M. Brunato and R. Battiti. RASH: A self-adaptive random search method. In Carlos Cotta, Marc Sevaux, and Kenneth Sörensen, editors, *Adaptive and Multilevel Metaheuristics*, volume 136 of *Studies in Computational Intelligence*. Springer, 2008.

[32] JB Butcher, David Verstraeten, Benjamin Schrauwen, CR Day, and PW Haycock. Reservoir computing and extreme learning machines for non-linear time-series data analysis. *Neural networks*, 38: 76–89, 2013.

[33] Paolo Campigotto, Andrea Passerini, and Roberto Battiti. Active learning of pareto fronts. *IEEE Transactions on Neural Networks and Learning Systems*, 25(3): 506–519, March 2014.

[34] Soumen Chakrabarti. *Mining the Web: discovering knowledge from hypertext data*. Morgan Kaufmann, 2003.

[35] O. Chapelle, M. Chi, and A. Zien. A continuation method for semi-supervised SVMs. In *Proceedings of the 23rd international conference on Machine learning*, page 192. ACM, 2006.

[36] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-supervised learning*. The MIT Press, Cambridge, MA, 2006.

[37] Olivier Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5): 1155–1178, 2007.

[38] Kevin J Cherkauer. Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks. In *Working notes of the AAAI workshop on integrating multiple learned models*, pages 15–21. Citeseer, 1996.

[39] Antonio Criminisi. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2-3): 81–227, 2011.

[40] R. Dawkins. *The selfish gene*. Oxford. Oxford University, 1976.

[41] R.F. Dell and M.H. Karwan. An interactive MCDM weight space reduction method utilizing a Tchebycheff utility function. *Naval Research Logistics*, 37(2): 403–418, 1990.

[42] J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Prentice Hall, Englewood Cliffs, NJ, 1983.

[43] Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.

[44] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *arXiv preprint cs/9501101*, 1995.

[45] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[46] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2): 337–407, 2000.

[47] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia: links, objects, time and space-structure in hypermedia systems: links, objects, time and space-structure in hypermedia systems*, pages 225–234. ACM, 1998.

[48] P.E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, 1981.

[49] F. Glover. Tabu search, Part I1. *ORSA journal on Computing*, 2(1): 4–32, 1990.

[50] F. Glover. Tabu search-Uncharted domains. *Annals of Operations Research*, 149(1): 89–98, 2007.

[51] F. Glover, M. Laguna, and R. Martí. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 39(3): 653–684, 2000.

[52] A. A. Goldstein. *Constructive Real Analysis*. Harper and Row, New York, 1967.

[53] Liyanaarachchi Lekamalage Chamara Kasun Guang-Bin Huang, Zuo Bai and Chi Man Vong. Local receptive fields based extreme learning machine. *IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE*, 10(2), 2015.

[54] Babak Hassibi, David G Stork, et al. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in neural information processing systems*, pages 164–164, 1993.

[55] J.A. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley Publishing Company, Inc., Redwood City, CA, 1991.

[56] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7): 1527–1554, 2006.

[57] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786): 504–507, 2006.

[58] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv: 1207.0580*, 2012.

[59] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8): 832–844, 1998.

[60] H. H. Hoos and T. Stuetzle. *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann, 2005.

[61] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Biophysics*, 79: 2554–2558, 1982.

[62] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5): 359–366, 1989.

[63] Gao Huang, Guang-Bin Huang, Shiji Song, and Keyou You. Trends in extreme learning machines: A review. *Neural Networks*, 61: 32–48, 2015.

[64] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: A new learning scheme of feedforward neural networks. In *Proceedings of International Joint Conference on Neural Networks (IJCNN2004)*, July 2004.

[65] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1): 489–501, 2006.

[66] F. Hutter, Y. Hamadi, H. Hoos, and K. Leyton-Brown. Performance prediction and automated tuning of randomized and parametric algorithms. *Principles and Practice of Constraint Programming-CP 2006*, pages 213–228, 2006.

[67] Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148: 34, 2001.

[68] Kevin Jarrett, Koray Kavukcuoglu, M Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *Proceedings of 2009 IEEE 12th International Conference on Computer Vision*, pages 2146–2153. IEEE, 2009.

[69] T. Joachims. Making large-scale SVM learning practical. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods-Support Vector Learning*, chapter 11. MIT-Press, Cambridge, Mass., 1999.

[70] Thorsten Joachims. Making large scale svm learning practical. Technical report, Universität Dortmund, 1999.

[71] Daniel Kahneman. *Thinking, fast and slow*. Farrar, Straus and Giroux, 2011.

[72] Y. Koren and L. Carmel. Robust linear dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 10(4): 459–470, 2004.

[73] N. Krasnogor and J. Smith. A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *IEEE Transactions on Evolutionary Computation*, 9(5): 474–488, Oct 2005.

[74] D.G. Krige. A statistical approach to some mine valuations and allied problems at the witwatersrand. Master's thesis, University of Witwatersrand, 1951.

[75] G. Lebanon. Metric learning for text documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 497–508, 2006.

[76] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361: 310, 1995.

[77] Yann LeCun, John S Denker, Sara A Solla, Richard E Howard, and Lawrence D Jackel. Optimal brain damage. In *NIPs*, volume 2, pages 598–605, 1989.

[78] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv: 1312.4400*, 2013.

[79] Mantas LukošEvičIus and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3): 127–149, 2009.

[80] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11): 2531–2560, 2002.

[81] M. Marchiori. The quest for correct information on the web: Hyper search engines. *Computer Networks and ISDN Systems*, 29(8-13): 1225–1235, 1997.

[82] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurrent Computation Program, C3P Report*, 826, 1989.

[83] E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. Technical Report AIM-1602, MIT Artificial Intelligence Laboratory and Center for Biological and Computational Learning, 1997.

[84] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanfor University, 1998.

[85] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. *arXiv preprint arXiv: 1211.5063*, 2012.

[86] Selcen (Pamuk) Phelps and Murat Köksalan. An interactive evolutionary metaheuristic for multiobjective combinatorial optimization. *Management Science*, 49(12): 1726–1738, 2003.

[87] John Platt et al. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods-support vector learning*, 3, 1999.

[88] William H Press. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[89] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1): 81–106, 1986.

[90] Frank Rosenblatt. *Principles of neurodynamics*. 1962.

[91] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*, volume 589. John Wiley & Sons, 2005.

[92] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D.E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations*. MIT Press, 1986.

[93] Craig Saunders, Alexander Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. In *(ICML-1998) Proceedings of the 15th International Conference on Machine Learning*, pages 515–521. Morgan Kaufmann, 1998.

[94] Andrew Saxe, Pang W Koh, Zhenghao Chen, Maneesh Bhand, Bipin Suresh, and Andrew Y Ng. On random weights and unsupervised feature learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1089–1096, 2011.

[95] Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2): 197–227, 1990.

[96] H. Scudder III. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3): 363–371, 1965.

[97] D. F. Shanno. Conjugate gradient methods with inexact searches. *Mathematics of Operations Research*, 3(3): 244–256, 1978.

[98] Joseph Sill, Gábor Takács, Lester Mackey, and David Lin. Feature-weighted linear stacking. *arXiv preprint arXiv: 0911.0460*, 2009.

[99] V. Sindhwani, S.S. Keerthi, and O. Chapelle. Deterministic annealing for semisupervised kernel machines. In *Proceedings of the 23rd international conference on Machine learning*, page 848. ACM, 2006.

[100] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. Technical Report NeuroCOLT

NC-TR-98-030, Royal Holloway College, University of London, UK, 1998.

[101] F. J. Solis and R. J-B. Wets. Minimization by random search techniques. *Mathematics of Operations Research*, 6(1): 19–30, February 1981.

[102] Ingo Steinwart, Don Hush, and Clint Scovel. Training svms without offset. *The Journal of Machine Learning Research*, 12: 141–202, 2011.

[103] R.E. Steuer and E. Choo. An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(1): 326–344, 1983.

[104] James Surowiecki. *The wisdom of crowds*. Random House Digital, Inc., 2005.

[105] Johan AK Suykens, Jos De Brabanter, Lukas Lukas, and Joos Vandewalle. Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, 48(1): 85–105, 2002.

[106] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3): 293–300, 1999.

[107] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[108] Kai Ming Ting and Ian H Witten. Issues in stacked generalization. *Journal of Artificial Intelligence Research*, 10: 271–289, 1999.

[109] Tony Van Gestel, Johan AK Suykens, Bart Baesens, Stijn Viaene, Jan Vanthienen, Guido Dedene, Bart De Moor, and Joos Vandewalle. Benchmarking least squares support vector machine classifiers. *Machine Learning*, 54(1): 5–32, 2004.

[110] Moshe Y Vardi. Is information technology destroying the middle class? *Communications of the ACM*, 58(2): 5–5, 2015.

[111] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 9999: 3371–3408, 2010.

[112] R. Watrous. Learning algorithms for connectionist networks: applied gradient methods of nonlinear optimization. Technical Report MS-CIS-87-51, Univ. of Penn, 1987.

[113] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10): 1550–1560, 1990.

[114] Bernard Widrow, Aaron Greenblatt, Youngsik Kim, and Dookun Park. The no-prop algorithm: A new learning algorithm for multilayer neural networks. *Neural Networks*, 37: 182–188, 2013.

[115] Bernard Widrow and Marcian E. Hoff. *Adaptive switching circuits*. Defense Technical Information Center, 1960.

[116] Bernard Widrow and Samuel D Stearns. *Adaptive signal processing*. Englewood Cliffs, NJ, Prentice-Hall, Inc., 491 p., 1, 1985.

[117] David H Wolpert. Stacked generalization. *Neural networks*, 5(2): 241–259, 1992.

[118] X. Yao. Evolving artificial neural networks. In *Proceedings of the IEEE*, 87(9): 1423–1447, 2002.

[119] X. Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2006.

[120] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. Semi-supervised learning using Gaussian fields and harmonic functions. In *International Conference on Machine Learning*, volume 3, pages 912–919, 2003.

[121] S. Zionts and J. Wallenius. An interactive multiple objective linear programming method for a class of underlying nonlinear utility functions. *Management Science*, 29(5): 519–529, 1983.

索　　引

BCO，参见脑–计算机优化
BFGS 法，200
bold driver（BP），82
CNN，参见神经网络，卷积神经网络
CoRSO，参见反馈搜索优化，合作反馈搜索优化
HITS，254
ILS，参见迭代局部搜索
K 均值算法，参见 K 均值
Kohonen 映射，参见自组织映射
K 均值，138
KISS 原则，205
LDA，参见线性判别分析
LS，参见局部搜索
LS-SVM，参见支持向量机，最小二乘支持向量机
logistic 回归，67，117
MA，参见模因算法
MLP，参见多层感知器神经网络
MOOP，参见多目标优化问题
MoRSO，参见反馈搜索优化，多目标反馈搜索优化
OSS，参见单步正割法
PCA，参见主成分分析
PageRank，251
RAS，参见反馈仿射振荡器
RNN，参见神经网络，递归神经网络
RSO，参见反馈搜索优化
SOM，参见自组织映射
SVM，参见支持向量机
TF-IDF，参见词频
Tichonov 正则化，36
Vapnik-Chervonenkis 维，97
VC 维，参见 Vapnik-Chervonenkis 维
VNS，参见可变邻域搜索
奥卡姆剃刀，38
百分位数，47
半监督学习，174
边干边学，216
标记，9
标记化，243
病态，192
测量误差，38
超限学习，128
池化层，91
重新开始，205，212
抽样搜索区域，203
储备池学习，121，128
词频，246
大峡谷特性，214
单步正割法，201
低密度分离假设，175
递归神经网络，122
调整搜索区域，203
迭代局部搜索，214
度量学习，180
堆叠，111
对偶二次规划，103
多层感知器神经网络，79
多分类器系统，113
多目标反馈搜索优化，232
多目标优化，233，235
多目标优化问题，232
少数服从多数，111
多样化，205，213，218
二分法，189
反馈（在优化阶段），235

反馈仿射振荡器，202，203，228
反馈禁忌策略，217
反馈搜索优化，212，215
反向传播法，80
方差缩减，111
非法二进制数，213
分类森林，113
分类问题，15
感知器，30
割线法，190
共轭梯度法，196
惯性振荡器，205
归一化距离，136，144
轨迹，213
过拟合，19，43
合作反馈搜索优化，222
核方法，101
互信息，64，65
回归问题，15
霍普菲尔德网络，124
机器学习，18
基尼混度，54
基于禁忌的反馈搜索优化，217
基于约束的聚类，180
激活函数，151
集成（ensemble），113
加权 K 近邻方法，12
加权主成分分析，160
加权最小二乘支持向量机，106
加性 logistic 回归，115
监督，9
监督学习，9
将用户作为学习的关键组件，215
交叉验证，19
交叉验证团体，113
焦点和上下文的可视化，168
杰卡德系数，248
结构风险，98
禁忌和多样化的基本关系，218
禁忌搜索，217
经验风险最小化，96
精确率，244
纠错码，115
局部极小，213
局部极小值，参见局部最优
局部加权回归，69
局部搜索，223
局部搜索过程的智能协作，222
局部最优，213
矩阵分解，260
卷积，90
卷积神经网络，90
决策森林，参见民主森林
决策森林中的特征排序，58
决策树，50
卡方，39
可变邻域搜索，214
课程学习，89
扩展存储器（memex），240
拉马克进化，224
拉普拉斯矩阵，159，177
懒惰，205
懒惰的初学者，10
离群值敏感性，160
李普希茨连续性，187
力控制，166
连续优化，222
量化误差，138，153
邻域，213
岭回归，35
流形假设，176
罗基奥方法，248
马氏距离，137，144
曼哈顿距离，136
民主森林，56
模因算法，223，224

内部表示与外部表示，135
脑–计算机优化，235
拟合与插值，38
凝聚聚类，142
牛顿定理，186
牛顿法，187，191
欧几里得距离，136
帕累托边界，234
帕累托最优，234
判别算法，18
批量反向传播法，81
平均值，47
平面国，155
评分矩阵，258
谱图绘制，169
去噪自动编码器，88
权威，254
缺失值，55
扰动，212
融合，111
软聚类，140
熵，53，64，65
社会学和政治学范式，222，224
深度神经网络，87
神经网络，76
神经元命名，153
生成方法，18
树状图，143
数据会向你坦白任何事情，14，20
双射策略，204
四分位数，47
搜索轨迹，213
搜索区域，224
搜索引擎，255
随机，82
缩减的卡方，43
泰勒级数，186
逃离某局部最小吸引域的最小禁忌值，220
特征，9
特征加权线性堆叠，112
特征选择，59
梯度下降，213
梯度下降法，80，194
提升法，113
条件熵，65
调整搜索区域，203
通过排列进行近似，249
统计学习理论，96
图形分布，168，171
退化的局部最优分布，171
椭球体变形，146
网页爬虫，241
网页挖掘，240
文档索引技术，242
沃罗诺伊图，139
无监督学习，134
无向加权图，165
吸引域，213，223
线性回归，27
线性判别分析，163
线性投影，156
相关比，63
相关系数，62
相似性度量，136
箱线图，47
向量–空间模型，245
协方差矩阵，145，157
协同推荐，258
信息增益，53
性能指标，244
修改和细化问题定义，235
学会飞翔，67
学习速率，151

训练，19
验证，19
隐式偏好与显式偏好，233
应力最小化，166
硬聚类，137
用户–项目矩阵，258
用户相似性或项目相似性，258
用于回归的支持向量，101
优化，185，211
余弦相似性，136
原型，135，151
噪声，38
召回率，244
正交投影，157
支持向量机，94，95
中位数，47
中心，255
主成分分析，158
装袋法，113
准确率–拒绝的折中，118
子抽样，91
自标记，175
自底向上聚类，参见凝聚聚类
自动编码器，86
自适应随机搜索，202
自组织映射，150
祖母细胞，150
最大化间隔，114
最近邻方法，11
最速下降，参见梯度下降
最小二乘法，40
最小二乘支持向量机，104



如今是人工智能高歌猛进的时代，机器学习的发展也如火如荼。然而，复杂的数学公式和难解的专业术语容易令刚接触这一领域的学习者望而生畏。有没有这样一本机器学习的书，能摒弃复杂的公式推导，带领读者通过实践来掌握机器学习的方法?

《机器学习与优化》正是这样一本书！它的写作脱胎于意大利特伦托大学机器学习与智能优化实验室（LION lab）的研究项目，语言轻松幽默，内容图文并茂，涵盖了机器学习中可能遇到的各方面知识。更重要的是，书中特别介绍了两个机器学习的应用，即信息检索和协同推荐，让读者在了解信息结构的同时，还能利用信息来预测相关的推荐项。

本书作者以及读者群发布的数据、指导说明和教学短片都可以在本书网站上找到：https://intelligent-optimization.org/LIONbook/。

本书内容要点：

- **监督学习**——线性模型、决策森林、神经网络、深度和卷积网络、支持向量机等
- **无监督模型和聚类**——K均值、自底而上聚类、自组织映射、谱图绘制、半监督学习等
- **优化是力量之源**——自动改进的局部方法、局部搜索和反馈搜索优化、合作反馈搜索优化、多目标反馈搜索优化等
- **应用精选**——文本和网页挖掘，电影的协同推荐系统

LIONlab
Trento University

图灵社区：iTuring.cn
热线：(010)51095186转600
分类建议 计算机/机器学习
人民邮电出版社网址：www.ptpress.com.cn

ISBN 978-7-115-48029-3
定价：89.00元

看完了

如果您对本书内容有疑问，可发邮件至 contact@turingbook.com，会有编辑或作译者协助答疑。也可访问图灵社区，参与本书讨论。

如果是有关电子书的建议或问题，请联系专用客服邮箱：
ebook@turingbook.com。

在这可以找到我们：

微博 @图灵教育：好书、活动每日播报
微博 @图灵社区：电子书和好文章的消息
微博 @图灵新知：图灵教育的科普小组
微信 图灵访谈：ituring_interview，讲述码农精彩人生
微信 图灵教育：turingbooks